**UNIVERSITY OF LILLE**
Graduate School No. 632: Engineering and System Sciences (ENGSYS)
Laboratory of Multiphysics and Multiscale Mechanics (LaMcube) – CNRS, UMR 9013

# Efficient Structure-Preserving Proper Generalized Decomposition Solvers for Structural Dynamics

submitted by **Clément VELLA**
for the degree of *Philosophiæ Doctor* in Mechanical Engineering

October 31st, 2024

| | |
|---|---|
| **Karen VEROY-GREPL** | Rapportrice & examinatrice<br>Full Professor, Technische Universiteit Eindhoven |
| **Amine AMMAR** | Rapporteur & examinateur<br>Professeur des Universités, Arts et Métiers ParisTech |
| **Anthony GRAVOUIL** | Président & examinateur<br>Professeur des Universités, INSA Lyon |
| **Diane GUIGNARD** | Examinatrice<br>Associate Professor, University of Ottawa |
| **Pedro DÍEZ** | Examinateur<br>Full Professor, Universitat Politècnica de Catalunya |
| **Pierre GOSSELET** | Encadrant de thèse & membre invité<br>CNRS Professor, Université de Lille |
| **Serge PRUDHOMME** | Co-Directeur de thèse<br>Full Professor, Polytechnique Montréal |
| **Abdelbacet OUESLATI** | Directeur de thèse<br>Maître de Conférences, Université de Lille |

**UNIVERSITÉ DE LILLE**

École Doctorale No. 632 : Sciences de l'Ingénierie et des Systèmes (ENGSYS)
Laboratoire de Mécanique, Multiphysique, Multiéchelle (LaMcube) – CNRS, UMR 9013

# Solveurs efficaces et préservant la structure par Décomposition Propre Généralisée pour la dynamique des solides

présenté par **Clément VELLA**
pour le diplôme de *Philosophiæ Doctor* en Génie Mécanique

31 Octobre 2024

| | |
|---|---|
| **Karen VEROY-GREPL** | Rapportrice & examinatrice<br>Full Professor, Technische Universiteit Eindhoven |
| **Amine AMMAR** | Rapporteur & examinateur<br>Professeur des Universités, Arts et Métiers ParisTech |
| **Anthony GRAVOUIL** | Président & examinateur<br>Professeur des Universités, INSA Lyon |
| **Diane GUIGNARD** | Examinatrice<br>Associate Professor, University of Ottawa |
| **Pedro DÍEZ** | Examinateur<br>Full Professor, Universitat Politècnica de Catalunya |
| **Pierre GOSSELET** | Encadrant de thèse & membre invité<br>CNRS Professor, Université de Lille |
| **Serge PRUDHOMME** | Co-Directeur de thèse<br>Full Professor, Polytechnique Montréal |
| **Abdelbacet OUESLATI** | Directeur de thèse<br>Maître de Conférences, Université de Lille |

## ACKNOWLEDGEMENTS

I would like to express my gratitude to the members of the jury, who kindly accepted to examine my work. I have learned a lot from their contributions, as much of my knowledge on the topic has been shaped by their work. I would like to thank Abdelbacet Oueslati, who trusted me to undertake this project. I am thankful to Serge Prudhomme for providing unfailing support, wise guidance, and freedom to explore. I am grateful to Pierre Gosselet for his significant contributions, thoughtful ideas, and strong support in the development of my work. I would like to acknowledge the financial support from the CDSN by the French Ministry of National Education, Higher Education, Research and Innovation.

I am deeply grateful to my family, particularly to my parents Marion and Pascal, my brother Julien, and my best friend and spouse Jhen, for their constant support and care throughout my life.

This thesis is dedicated to everyone aforementioned.

# ABSTRACT

The increasing complexity of engineering designs, particularly in sectors like construction, automotive, and aerospace, demands efficient computational methods to simulate dynamic behaviors. In particular, considerable computational burden arises from the dependence of the solution fields on both the space and time variables. As new design spaces are explored, including variations in geometry and mass distributions, the need to accurately model dynamic effects, such as vibration and damping, becomes critical.

This thesis proposes Reduced-Order Modeling techniques, focusing on computational efficiency, structure preservation, and the ability to parametrize the reduced model with respect to damping parameters. The Proper Generalized Decomposition (PGD) method is first revisited using the Hamiltonian formalism and then extended to enable the construction of parameterized reduced models for viscoelastic materials under transient loads. Initially, a displacement-momentum formulation was introduced to enhance robustness. Further, the PGD solver was adapted in a space-discrete, time-continuous Hamiltonian framework, to tackle the preservation of the symplectic structure with respect to the time variable. An approximation inspired by Modal Analysis was formulated to build the reduced basis efficiently, complemented by convergence acceleration techniques. Finally, the PGD framework was extended to allow for efficient parametrization of the reduced model using Rayleigh damping parameters. The PGD reduced basis was subsequently employed as a surrogate to determine optimal damping coefficients with respect to a given snapshot using the Particle Swarm Optimization algorithm. Several numerical experiments demonstrate the performance of the proposed methodology.

Keywords: Reduced-Order Modeling, Proper Generalized Decomposition, Structural Dynamics, Iterative Solver, Symplectic Structure, Damped Surrogate.

# RÉSUMÉ

La complexité croissante des conceptions en ingénierie, notamment dans des secteurs tels que la construction, l'automobile et l'aérospatiale, exige des méthodes de calcul efficaces pour simuler les comportements dynamiques. En particulier, la simple dépendance des champs solution par rapport aux variables d'espace et de temps engendre une charge de calcul considérable. Alors que de nouveaux espaces de conception sont explorés, incluant des variations dans la géométrie et la répartition des masses, la nécessité de modéliser avec précision les effets dynamiques, tels que les vibrations et l'amortissement, devient cruciale.

Cette thèse propose des techniques de Réduction de Modèle, avec pour objectifs l'efficacité du calcul, la préservation des structures mathématiques et la capacité à paramétrer le modèle réduit vis-à-vis de l'amortissement. La méthode de Décomposition Propre Généralisée (PGD) est d'abord revisitée en utilisant le formalisme hamiltonien, puis étendue pour permettre la construction de modèles réduits paramétrés pour les matériaux viscoélastiques soumis à des chargements transitoires. Initialement, une formulation déplacement-moment conjugué a été introduite afin d'améliorer la robustesse. Puis, le solveur PGD a été adapté dans un cadre hamiltonien discrétisé en espace et continu en temps, pour traiter la préservation de la structure symplectique par rapport au temps. Une approximation inspirée de l'Analyse Modale a été formulée pour construire efficacement la base réduite, complétée par des techniques d'accélération de convergence. Enfin, le solveur PGD a été étendu pour inclure les coefficients d'amortissement de Rayleigh comme variables du modèle réduit. La base réduite PGD est ensuite utilisée comme méta-modèle par un algorithme d'Optimisation par Essaim Particulaire pour déterminer des coefficients d'amortissement optimaux à partir d'un échantillon donné. Plusieurs expériences numériques illustrent les performances de la méthodologie proposée.

Mots-clefs: Réduction de modèle, Décomposition Propre Généralisée, Dynamique des Structures, solveur itératif, structure symplectique, meta-modèle amorti.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii

ABSTRACT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv

RÉSUMÉ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . v

TABLE OF CONTENTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vi

LIST OF TABLES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix

LIST OF FIGURES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x

LIST OF APPENDICES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xiv

CHAPTER 1 INTRODUCTION . . . . . . . . . . . . . . . . . . . . . . . . . . 1
1.1 Computational Mechanics and scope of the thesis . . . . . . . . . . . . . 1
1.2 Reduced-Order Modeling . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.2.1 Context and stakes with a playful analogy . . . . . . . . . . . . . . 4
1.2.2 Formal context in linear elastodynamics . . . . . . . . . . . . . . . 6
1.3 Literature review . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
1.3.1 Modal Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
1.3.2 Dynamic Substructuring . . . . . . . . . . . . . . . . . . . . . . . . 13
1.3.3 Proper Orthogonal Decomposition . . . . . . . . . . . . . . . . . . . 14
1.3.4 Proper Symplectic Decomposition . . . . . . . . . . . . . . . . . . . 15
1.3.5 Proper Generalized Decomposition . . . . . . . . . . . . . . . . . . 18
1.3.6 Viscous damping in ROM for structural dynamics . . . . . . . . . . . 22
1.4 Thesis objective and contributions . . . . . . . . . . . . . . . . . . . . 23

CHAPTER 2 ON A GALERKIN-BASED PROPER GENERALIZED DECOMPOSITION FOR THE 1D WAVE EQUATION: DEVELOPMENT OF A ROBUST DISPLACEMENT-MOMENTUM APPROACH . . . . . . . . . . . . . . . . . . . . 27
2.1 Model problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.1.1 Strong formulation . . . . . . . . . . . . . . . . . . . . . . . . . . 28
2.1.2 Analytical solution . . . . . . . . . . . . . . . . . . . . . . . . . . 29
2.2 Weak formulations of the problem . . . . . . . . . . . . . . . . . . . . . 30

2.2.1 Hamilton's Weak Principle . . . 31
2.2.2 The Lagrangian formalism . . . 31
2.2.3 The Hamiltonian formalism . . . 36
2.3 PGD reduced-order modeling . . . 40
2.3.1 Lagrangian-based PGD . . . 41
2.3.2 Hamiltonian-based PGD . . . 43
2.3.3 Updating procedure of the temporal modes and Gram-Schmidt process 46
2.3.4 Adaptive fixed-point algorithm . . . 49
2.4 Numerical results and discussion . . . 51
2.4.1 Test cases . . . 51
2.4.2 Comparison method and performance criteria . . . 53
2.4.3 Case 1: Neumann BC without updating procedure . . . 54
2.4.4 Case 2: Neumann BC with updating procedure . . . 56
2.4.5 Case 3: Oscillating Dirichlet BC . . . 62
2.4.6 Case 4: Comparison with analytical solution . . . 64
2.4.7 Case 5: Damped bar with Neumann BC . . . 67
2.4.8 Further discussion . . . 70
2.5 Conclusion . . . 71

CHAPTER 3 ON AN EFFICIENT, SYMPLECTICITY-PRESERVING SPACE-TIME PROPER GENERALIZED DECOMPOSITION . . . 72
3.1 Model problem . . . 73
3.1.1 Strong formulation . . . 73
3.1.2 Semi-weak formulation . . . 74
3.1.3 Spatial discretization . . . 75
3.2 The Hamiltonian formalism . . . 76
3.2.1 Hamilton's Weak Principle . . . 76
3.2.2 Symplectic structure . . . 78
3.2.3 Discretization in time of the Hamiltonian problem . . . 80
3.3 PGD reduced-order modeling . . . 80
3.3.1 Fixed-point strategy . . . 81
3.3.2 Aitken acceleration . . . 85
3.3.3 Temporal update and symplectic structure . . . 88
3.3.4 Projection in Ritz subspace . . . 91
3.4 Numerical examples and discussion . . . 93
3.4.1 Test case: asymmetric triangle wave Neumann boundary condition . . . 93

3.4.2 Comparison method and performance criteria . . . . . . . . . . . . . . 97
3.4.3 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
3.4.4 Further discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
3.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104

CHAPTER 4 EXTENSION OF THE PROPER GENERALIZED DECOMPOSITION FOR SURROGATE MODELING WITH APPLICATION TO THE IDENTIFICATION OF RAYLEIGH DAMPING PARAMETERS . . . . . . . . . . . . . . . . . . . 105
4.1 Model problem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
4.1.1 Strong formulation . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
4.1.2 Semi-weak formulation . . . . . . . . . . . . . . . . . . . . . . . . . 107
4.1.3 Spatial discretization . . . . . . . . . . . . . . . . . . . . . . . . . . 108
4.1.4 Damping modeling and parameter discretization . . . . . . . . . . . . 109
4.1.5 Discretization in time . . . . . . . . . . . . . . . . . . . . . . . . . . 110
4.2 Parametric PGD reduced-order modeling . . . . . . . . . . . . . . . . . 112
4.2.1 Fixed-point algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 114
4.2.2 Problem in space . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116
4.3 Parameter identification using Particle Swarm Optimization . . . . . . . . . 120
4.4 Numerical examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
4.4.1 Test case: parameter optimization with respect to a snapshot . . . . . . 123
4.4.2 Comparison method and performance criteria . . . . . . . . . . . . . . 124
4.4.3 Numerical results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 126
4.4.4 Further discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134

CHAPTER 5 CONCLUSION . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
5.1 Summary of Works . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
5.2 Limitations and Future Research . . . . . . . . . . . . . . . . . . . . . . 136

REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139

APPENDICES . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150

# LIST OF TABLES

Table 3.1 Time efficiency of the reduced-order models (SVD, PGD LU, PGD Ritz) with respect to different spatial discretizations and PGD Ritz speedup compared to other methods. . . . . . . . . . . . . . . . . . . . . . . . . . 101

Table 4.1 Time efficiency of the PGD offline phase with respect to different spatial discretizations with $n_\alpha = n_\beta = 190$. . . . . . . . . . . . . . . . . . . . . . 129

Table 4.2 Time efficiency of the FEM solver described in Algorithm 4 for one computation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129

Table 4.3 Summary of the time and memory efficiency of the developed PGD solver against those of a conventional FEM for a computation with $2n = 3{,}002{,}406$ spatial DOFs, $n_t = 4{,}800$ time-steps and $n_\alpha = n_\beta = 190$ values for the damping parameters (the PGD results are provided by the code while the FEM results are estimated based on the result for a single computation). . . . . . . . . . . . . . . . . . . . . . . . . . . 134

# LIST OF FIGURES

Figure 1.1 Venn diagram of the disciplines that made Computational Mechanics. . 2

Figure 1.2 Three levels of representation (from left to right): reduced-order pumpkin, assembled pumpkin [10] and a real pumpkin [11]. . . . . . . . . . . 5

Figure 2.1 Case 1. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3. . . . . . 55

Figure 2.2 Case 1. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field. . . . . . . . 56

Figure 2.3 Case 1. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time. . . . . . 57

Figure 2.4 Case 2. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3. . . . . . 58

Figure 2.5 Case 2. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field. . . . . . . . 59

Figure 2.6 Case 2. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time. . . . . . 60

Figure 2.7 Case 3. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3. . . . . . 61

Figure 2.8 Case 3. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field. . . . . . . . 62

Figure 2.9 Case 3. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time. . . . . . 63

Figure 2.10 Case 4. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3. . . . . . 64

Figure 2.11 Case 4. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field. . . . . . . . 65

Figure 2.12 Case 4. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time. . . . . . 66

Figure 2.13 Case 5. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3. . . . . . 67

Figure 2.14 Case 5. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field. . . . . . . . 68

Figure 2.15 Case 5. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time. . . . . . 69

Figure 3.1 Scheme of the test case. . . . . . . . . . . . . . . . . . . . . . . . . . 94

Figure 3.2 Evolution in time of the boundary traction $g_N \cdot n$. . . . . . . . . . . . . 94

Figure 3.3 Number of iterations for 20 modes without and with Aitken acceleration 95

Figure 3.4 Visualization [95] of the first three temporal modes (normalized) with and without Aiken acceleration, herein denoted $\widetilde{\psi}_i^q$ and $\psi_i^q$, respectively. 96

Figure 3.5 (Left) Error between the reference solutions and the SVD, PGD or Modal Decomposition approximations for 244,926 spatial DOF with 50 modes. (Right) Error between the reference solutions and the SVD, PGD or Modal Decomposition approximations for 36,774 spatial DOF with 300 modes. ($y$-axis has log scale) . . . . . . . . . . . . . . . . . . 98

Figure 3.6 Real execution time for the full-order model (FEM) and the reduced-order models (SVD, PGD LU, PGD Ritz) with respect to different spatial discretizations ($y$-axis has log scale). . . . . . . . . . . . . . . . 99

Figure 3.7 Detail on the real execution time for the full-order model (FEM) and the reduced-order models (PGD LU and PGD Ritz) with respect to different space discretizations ($y$-axis has log scale). . . . . . . . . . . . 100

Figure 3.8 Visualization of the first three spatial modes (normalized) for the Modal Decomposition, PGD LU and PGD Ritz on the first, second and third columns, respectively and undeformed configuration in low opacity. . . 102

Figure 4.1 Flowchart of the conventional fixed-point algorithm with associated computational complexities. . . . . . . . . . . . . . . . . . . . . . . . . 116

Figure 4.2 Flowchart of the PGD offline phase with the hybrid space solver. . . . . 119

Figure 4.3 Pseudo flowchart of the deployed solution. . . . . . . . . . . . . . . . . . . . . . 121

Figure 4.4 Schematic of the 3D beam $\bar{\Omega} = [0,6] \times [0,1] \times [0,1]$. . . . . . . . . . . . 123

Figure 4.5 Schematic of the boundary multiaxial load $\sigma \cdot n$ prescribed on $\partial\Omega_N$. . . 125

Figure 4.6 1st spatial mode. . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.7 2nd spatial mode. . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.8 3rd spatial mode. . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.9 4th spatial mode. . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.10 First four temporal modes. . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.11 First four $\alpha$-modes. . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.12 First four $\beta$-modes. . . . . . . . . . . . . . . . . . . . . . . . . . 127

Figure 4.13 (Left) Relative error $\epsilon_{ROM}$ between the reference fields and the PGD fields for the three strategies for the problem in space ($y$-axis in log scale). (Right) Factor $w_i$ of the $i^{\text{th}}$ enrichment for the three strategies for the problem in space ($y$-axis in log scale). . . . . . . . . . . . . . . 129

Figure 4.14 Impact of an initial guess on the number of MinRes iterations (cumulative sum of MinRes iterations per fixed-point iteration). . . . . . . . . 130

Figure 4.15 (Left) Time distribution of the developed PGD solver described on Figure 4.2 ($2n = 3{,}002{,}406$ spatial DOFs, $n_t = 4{,}800$ time-steps and $n_\alpha = n_\beta = 190$ values for the damping parameters). (Right) Time distribution of the FEM solver described in Algorithm 4 ($2n = 3{,}002{,}406$ spatial DOFs, $n_t = 4{,}800$ time-steps for one computation). . . . . . . . 131

Figure 4.16 (Left) Values of the global best cost $J^k_{p_b}$ and the global worst cost $J^k_{p_w}$ during the PSO iterations. (Right) Values of the errors $\epsilon_\alpha$ and $\epsilon_\beta$ for the global best particle $p_b$ during the PSO iterations. . . . . . . . . . . 132

Figure 4.17 The swarm particles throughout PSO iterations. . . . . . . . . . . . . . 133

Figure 4.18 Cost function $J$ over $\mathcal{D}$. . . . . . . . . . . . . . . . . . . . . . . . . . . 133

# LIST OF APPENDICES

Appendix A Time operators . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
Appendix B Definitions of the parametric PGD subproblems . . . . . . . . . . . . 151

# CHAPTER 1 INTRODUCTION

## 1.1 Computational Mechanics and scope of the thesis

The last half-century has seen the emergence of computing hardware as a medium for scientific experimentation. Before then, except for a few attempts to automate computations, computers were none other than human beings themselves. Mathematical and modeling tools already had well established foundations. As a matter of fact, all the formulations in this manuscript are based on variational calculus, from which the Principle of Least Action stems, and were known at the time. It was also common to use numerical approximations to solve initial and boundary-value problems that were not analytically solvable. Astronomy is undoubtedly a remarkable application of these methods, as it has never ceased to fascinate humankind throughout the ages and across civilizations. It was that same fascination that led to space exploration and sparked off the computational era. Indeed, if approximation methods were well mastered, human computers faced a problem of efficiency in achieving ever finer approximations. In particular, trajectory calculation was an essential work at the National Aeronautics and Space Administration (NASA). A too coarse approximation inevitably led to dramatic deviations, which cannot be allowed especially for inhabited flights. Approximation refinement required to perform so many operations that tasks exceeded human capabilities. In 1959, NASA acquired the IBM 7090 computer. It was capable of 24 thousand multiplications per second. This computing power was phenomenal at the time, but there was still the challenge of figuring out how to operate the computer. It required a very specific set of skills: programs were written in languages such as Fortran or COBOL, the source code was punched onto cards using keypunch machines, then cards were fed into the card reader in a specific order to schedule the job for execution. Subsequently, outputs were collected for analysis. This laborious process was inherently prone to error, so much that most engineers and astronauts distrusted it [1]. A lot of effort was put into making these tools reliable, which indisputably contributed to make the space program a success. Essentially, hardware computers provided engineers with the necessary computing power to numerically experiment and predict flight missions that could not be tested otherwise. After many years of progress, the mindset regarding numerical simulation changed radically: it would nowadays be inconceivable not to numerically experiment before manufacturing and proceed to real testing [2]. It is the result of a joint effort between the mathematics, computer science and mechanical engineering communities. Eventually, this fruitful collaboration gave birth to the discipline commonly referred to as Computational Mechanics (CM), a term coined by J. T. Oden (1936 – 2023), and later as Computational Sciences and Engineering.

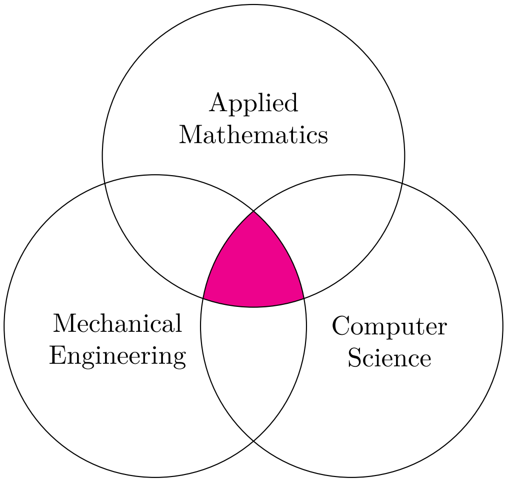


Figure 1.1 Venn diagram of the disciplines that made Computational Mechanics.

Among the immense variety of tools that are now available, Finite Element (FE) modeling has stood out as a tried-and-tested method. The FE Method (FEM) is particularly appreciated for its error convergence properties, versatility and automation, although none of these are without fault. For instance, many open-source and commercial FE packages or software have been successfully developed over the years: NASTRAN, Code_Aster [3], deal.II [4], FEniCS [5], FreeFEM [6], to cite a few. The sub-fields of Validation and Verification also played a crucial role in the adoption of numerical tools [2]. Validation is focused on whether the equations used to model a phenomenon are suitable. In other words, the predictions of the model are compared with real observations and experimental measurements [2]. It addresses the question: *are we solving the right equations?* On the other hand, Verification aims to check the quality of the numerical result against the solutions to the model equations, which one solves in an approximated manner, e.g. considering a FE approximation. It involves the construction of error estimators that quantitatively measure the quality of the approached solution [7]. It answers the question: *are we solving the equations right?*

Not only CM has provided predictive tools to tackle existing problems but it has also opened an avenue to explore novel, more complex models. As a result, the complexity of the problems that are tackled have substantially increased, e.g. in design, control, optimization and uncertainty quantification [8]. Accounting for the variation of parameters that govern simulations has become essential, driven by the need to finely study physical behaviors and improve the predictive capacity of computational tools. These include, for example, variations in boundary or initial conditions, material and geometric parameters. It has led to a drastic increase in the number of calculations to be performed, due to the so-called curse of dimensionality [9]. Despite advances in computing hardware, conventional CM methods have become obsolete to deal with the demanding simulations mentioned above. Yet again, reaching the limits of simulation capabilities has motivated a surge of interest from the scientific community. Model reduction methods have emerged in this context, introducing innovative strategies tailored to solve such high-dimensional problems.

The mere dependence of the fields on both the space and time variables is a source of computational burden to simulate transient phenomena. Applications in structural dynamics often require fast methods to efficiently estimate solutions for real-time simulations, digital twinning, uncertainty quantification analyses or multi-query optimization. In mechanical engineering, e.g. in the construction, automotive and aerospace industries, advancements in materials and manufacturing have led to optimized designs with reduced masses. This effort is driven by environmental demands. In the long run, lighter structures require less material and energy to be manufactured, and in the case of transportation, energy consumption in operation would be decreased as well. In transportation, as well as in energy production or storage, the quest for performance can lead to the exploration of new rotor regimes for engines, water or wind turbines. All these innovations give a major importance to dynamic effects. Furthermore, damping control is a fundamental issue for the performance of vibrating structures. The application of model reduction techniques to transient dissipative behaviors would enable the automatic construction of damped meta-models, hence improving the predictive capabilities of numerical simulation. Thus, in addition to space and time, model parameters related to damping may be considered as variables, which substantially increases the difficulty of the computation.

Over the last decade, Reduced-Order Modeling for structural dynamics has significantly progressed. Traditional methods like modal superposition and dynamic substructuring remain key in engineering for reducing computational costs. Data-driven methods, such as the Proper Orthogonal Decomposition, have achieved significant advancements in efficiently capturing system dynamics by identifying the dominant modes from snapshots. Reduced-Order Models (ROM) dedicated to preserve the symplectic structure have generated interest for problems that can be described in terms of the canonical Hamilton's equations. The preservation of this feature by the ROM is guaranteed by Proper Symplectic Decomposition methods and is especially relevant for accurate long-term predictions. Additionally, Proper Generalized Decomposition provides a versatile framework for constructing parameterized ROMs with respect to model parameters such as material properties, geometry, and boundary conditions, to cite a few. These advancements are supported by a substantial body of literature and have been applied across various fields in engineering sciences.

Despite significant advancements, ROM for structural dynamics still faces several challenges. One major shortcoming is the limited ability of many ROMs to accurately capture nonlinear and non-stationary behavior, which is common in real-world applications. Most models also struggle with scalability, particularly when applied to large-scale or highly complex systems. Reduction always involves a trade-off between accuracy and computational efficiency, making it difficult to develop models that are both precise and fast. Another issue is the lack of robust error estimation and adaptive refinement techniques, which are crucial for ensuring reliability across diverse applications. Finally, the integration of ROMs into existing industrial workflows can be problematic due to the intrusive nature of some techniques, compatibility issues and the need for specialized knowledge to implement these advanced methods effectively.

The general objective of this thesis is to propose novel ROM implementations that specifically improve robustness, scalability, and enable parametrization with respect to damping model parameters. Chapter 1 introduces thereafter some fundamental principles related to Reduced-Order Modeling, as well as a literature review focused on its applications to structural dynamics. The main objective and specific objectives of the thesis will be stated at the end of this chapter. Chapter 2 presents the initial implementation of the proposed ROM, based on the Proper Generalized Decomposition, and highlights improvements in robustness compared to previously published methods. It includes a specific study on the operator's condition numbers and the use of different metrics for the orthogonalization of the computed modes. Chapter 3 focuses on enhancing scalability and offers an interpretation of the ROM in terms of the canonical Hamilton's equations. This chapter also proposes a PGD framework that is adapted to the preservation of the symplectic structure with respect to the time variable. Chapter 4 incorporates parametric damping and explores the use of surrogate modeling for damping optimization purposes. Finally, Chapter 5 provides concluding remarks, discusses the limitations of the current work, and outlines potential directions for future research.

## 1.2 Reduced-Order Modeling

### 1.2.1 Context and stakes with a playful analogy

The underlying idea of Reduced-Order Modeling (ROM[1]) is that a volume of information, whether finite or infinite, structured or unstructured, can be approximately represented using a finite set of elementary patterns. From a storage perspective, a reduced model is considered successful if its set of elementary patterns are contained in a much smaller volume than that of

[1] In this thesis, ROM will refer to either Reduced-Order Modeling or Reduced-Order Model

the original, full-order information. Moreover, its proficiency is assessed based on how accurate it can represent the original data. The most widespread application is file compression. For example, `.zip` and `.png` are formats that support lossless data compression. They are optimal in the way they provide lighter files without any loss of information. Alternatively, the `.jpeg` file format allows for lighter files with lossy compression techniques. It will cause a loss of information, characterized by the presence of artifacts in the compressed image files. The main point is that the image may become pixelated, yet its content is still recognizable. Toy building blocks illustrate well a lossless reduction process (see Figure 1.2). The assembled pumpkin is made of 102 building blocks. However, the assembly can be summed up by no more than 11 unique classes of blocks. It essentially means that the knowledge of 11 building blocks is sufficient to have a complete understanding of the assembly. This would represent about a 10% memory gain. Furthermore, it is worth noting the similar shapes and colors between the assembled pumpkin and the building blocks. It highlights feature preservation between the full-order and reduced-order representations. The meaning of structure preservation in physics is an essential topic in ROM and will be discussed later. The question of the uniqueness of the reduced model will be tackled as well.

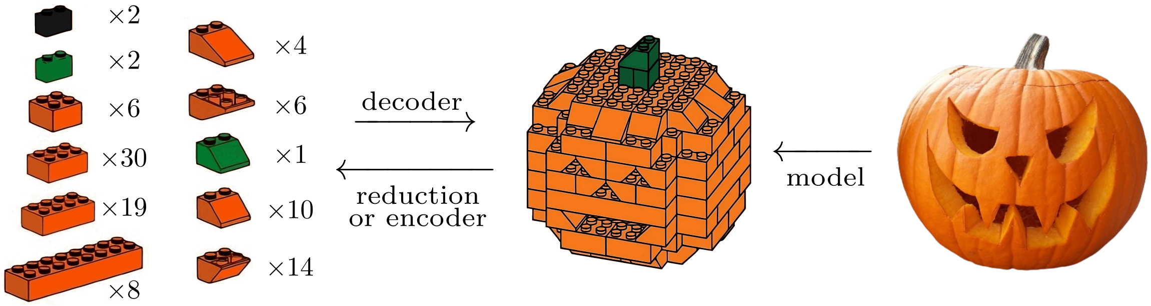


Figure 1.2 Three levels of representation (from left to right): reduced-order pumpkin, assembled pumpkin [10] and a real pumpkin [11].

Computing complexity is another crucial issue in ROM. Two procedures must be defined:

1. Reduction or encoding: from the full-order representation, find the elementary patterns that capture most or all the given information. This stage is commonly referred to as the offline phase;

2. Decoding, i.e. the inverse operation: from the elementary patterns, reconstruct an approximation of the full-order representation. This is the online phase.

It is clear that the second operation is the easiest: given the block components and user instructions, it is rather easy to build the pumpkin. For instance, in the case of image compression, one needs the appropriate piece of code or reader to open the compressed file. Strictly speaking, when one opens an image, the operating system calls the appropriate algorithm to decode the compressed file and displays it. Conversely, it is not trivial to determine a minimal set of blocks that constitutes the assembled pumpkin or an image. Whether the offline and online phases can be performed efficiently determines the success of the ROM as well.

Furthermore, the diagram on the right of Figure 1.2 completes the analogy with the relation between a real-world observation and a model. It can be considered that the assembled pumpkin models a real pumpkin. Again, features that are essential such as colors, overall shape and carvings are preserved by the model. The other way around, given the assembled pumpkin, one would eventually be able to tell it is meant to represent a pumpkin – although this may be contingent to social and cultural influences (the decoder). The key ideas are as follows: the information is abstracted and reduced from the right to the left with losses along the way; yet, if the major features were purposely preserved and given the proper decoder, it will be possible to use the left-most representation to infer characteristics of the real object. Furthermore, it is essential to explain how these concepts transpose when it comes to describe physics.

### 1.2.2 Formal context in linear elastodynamics

In the context of this thesis, we shall consider the model of linear elastodynamics in $d$ spatial dimensions under the assumption of infinitesimal deformation ($d = 1, 2,$ or $3$). Let $\Omega$ be an open bounded subset of $\mathbb{R}^d$, with Lipschitz boundary $\partial\Omega$, and let $\mathcal{I} = (0, T)$ denote the time interval. The boundary $\partial\Omega$ is supposed to be decomposed into two portions, $\partial\Omega_D$ and $\partial\Omega_N$, such that $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. The displacement field $u : \bar{\Omega} \times \bar{\mathcal{I}} \to \mathbb{R}^d$ satisfies the following partial differential equation[2]:

$$\rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma(u) = f, \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{1.1}$$

where, in the case of infinitesimal deformation, the stress tensor $\sigma(u)$ and strain tensor $\varepsilon(u)$ are given by:

$$\sigma(u) = \mathbb{E} : \varepsilon(u), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{1.2}$$

$$\varepsilon(u) = \frac{1}{2}\Big(\nabla u + \big(\nabla u\big)^T\Big), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{1.3}$$

[2] Parameters others than space and time are not considered as variable here for the sake of clarity in the presentation. Extra-parameter dependence will be tackled in Chapter 4.

and is subjected to the initial conditions:

$$u(x,0) = u_0(x), \qquad \forall x \in \Omega, \tag{1.4}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \quad \forall x \in \Omega, \tag{1.5}$$

as well as to the boundary conditions:

$$u(x,t) = 0, \qquad\qquad \forall (x,t) \in \partial\Omega_D \times \mathcal{I}, \tag{1.6}$$

$$\sigma(u) \cdot n = g_N(x,t), \quad \forall (x,t) \in \partial\Omega_N \times \mathcal{I}. \tag{1.7}$$

The functions $f : \Omega \times \mathcal{I} \to \mathbb{R}^d$, $u_0 : \Omega \to \mathbb{R}^d$, $v_0 : \Omega \to \mathbb{R}^d$, and $g_N : \partial\Omega_N \times \mathcal{I} \to \mathbb{R}^d$ are supposed to be sufficiently regular to yield a well-posed problem. Equations (1.1) to (1.7) constitute the model, our main goal being the description of the displacement field $u$ on $\bar{\Omega} \times \bar{\mathcal{I}}$. The medium occupied by $\bar{\Omega}$ is assumed to have local material properties described by the tensor of elasticity $\mathbb{E}$ and density $\rho$. Moreover, $\mathbb{E}$ is assumed to be symmetric. In the following, we will denote the first and second time derivatives by $\dot{u} = \partial u/\partial t$ and $\ddot{u} = \partial^2 u/\partial t^2$.

> **Key point** Thereby, the information of interest is that of the displacement field $u$ (and potentially some of its time derivatives). The purpose is twofold, as we want both to compress the representation of $u$ and to reduce the computational complexity associated with the model simulation.

Further, we consider here the semi-weak Finite Element formulation with respect to the spatial variable. A thorough presentation of this development will be given in the subsequent chapters. For now, let us admit the following semi-weak formulation of the problem:

Find $u = u(\cdot, t) \in V$, for all $t \in \bar{\mathcal{I}}$, such that:

$$\int_\Omega \rho \ddot{u} \cdot u^* + \varepsilon(u) : \mathbb{E} : \varepsilon(u^*) \; dx = \int_\Omega f \cdot u^* \; dx + \int_{\partial\Omega_N} g_N \cdot u^* \; dx, \quad \forall u^* \in V, \quad \forall t \in \mathcal{I}, \tag{1.8}$$

and:

$$u(x,0) = u_0(x), \qquad \forall x \in \Omega, \tag{1.9}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \qquad \forall x \in \Omega, \tag{1.10}$$

where $V$ is the vector space of vector-valued functions defined on $\Omega$:

$$V = \left\{ v \in \left[ H^1(\Omega) \right]^d : \; v = 0 \text{ on } \partial\Omega_D \right\}.$$

> **Remark** There are many ways to obtain a semi-weak formulation (not unique) for the strong formulation stated above, such as the Principle of Virtual Work or the Principle of Least Action. The latter principles are mathematically equivalent, yet we will see that they lead to different semi-weak formulations that present different advantages and drawbacks.

The domain is partitioned into $N_e$ elements, denoted by $K_e$, such that $\overline{\Omega} = \cup_{e=1}^{N_e} K_e$ and $\mathrm{Int}(K_i) \cap \mathrm{Int}(K_j) = \varnothing$, $\forall i, j = 1, \dots, N_e$, $i \neq j$. We then associate with the mesh the finite-dimensional Finite Element space $W^h$, with $\dim(W^h) = s$, of scalar-valued continuous and piecewise polynomial functions defined on $\Omega$, that is:

$$W^h = \left\{ v_h : \Omega \to \mathbb{R} : \; v_h|_{K_e} \in \mathbb{P}_k(K_e), \; e = 1, \dots, N_e \right\},$$

where $\mathbb{P}_k(K_e)$ denotes the space of polynomial functions of degree $k$ on $K_e$. Let $\phi_i$, with $i = 1, \dots, s$, denote the basis functions of $W^h$, i.e. $W^h = \mathrm{span}\{\phi_i\}$. We then introduce the finite element subspace $V^h$ of $V$ such that:

$$V^h = \left[ W^h \right]^d \cap V,$$

and search for finite element solutions $u_h = u_h(\cdot, t) \in V^h$, $\forall t \in \bar{\mathcal{I}}$, in the form:

$$u_h(x, t) = \sum_{i=1}^{s} \phi_i(x) q_i(t),$$

where the vectors of degrees of freedom (DOFs), $q_j \in \mathbb{R}^d$, depend on time. The above problem can be conveniently recast in compact form as[3]:

$$M \ddot{\boldsymbol{q}}(t) + K \boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{1.11}$$

$$\boldsymbol{q}(0) = \boldsymbol{u}_0, \tag{1.12}$$

$$\dot{\boldsymbol{q}}(0) = \boldsymbol{v}_0, \tag{1.13}$$

where $M \in \mathbb{R}^{n \times n}$ and $K \in \mathbb{R}^{n \times n}$ are the global mass and stiffness matrices, respectively, both being symmetric and positive definite. $\boldsymbol{f}(t)$ is the load vector at time $t$, $\boldsymbol{q}(t)$ is the global vector of DOFs and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial vectors. The integer $n$ is the number of global DOFs in space, that is $n = d \times s$. Let us consider for now that $d = 1$. The finite element

[3] The development leading to this form is omitted for the sake of conciseness. The corresponding details can be found in Sections 2.2.2, 3.1.3 and 4.1.3 of the following chapters.

solution can be rewritten as:

$$u_h(x,t) = \boldsymbol{\phi}_h(x)\boldsymbol{q}(t),$$

with $\boldsymbol{\phi}_h$ such that:

$$\boldsymbol{\phi}_h(x) = \begin{bmatrix}\phi_1(x) & \phi_2(x) & \dots & \phi_n(x)\end{bmatrix}.$$

The development presented here is a classic FE formulation. It is interesting to interpret it as a kind of model reduction. Indeed, in the space-time continuous strong formulation, a solution $u$ is sought for in $V$, whose dimension is infinite. The FE formulation proposes to search for an approximation of $u$, denoted by $u_h$, in $V^h$, whose dimension is finite. Given a mesh, the basis functions $\phi_i$ are known and the $q_i$ are the time-dependent unknowns of the problem. The basis functions are defined on $\Omega$ but are typically chosen with a small support, i.e. they are non-zero only locally, more precisely, $\phi_i$ is non-zero only in the patch of elements that share node $i$[4]. Therefore, it is intuitively understood that each DOF carries a rather small amount of information regarding $u_h$. Moreover, the quality of the FEM depends directly on the mesh size, as the errors should converge to zero as it goes to zero[5]. Finer meshes lead to a high number of basis functions. For that reason, the FEM cannot be truly considered as a ROM.

The goal of ROM is to find a new set of basis functions, which could be defined in terms of the FE basis functions. The combined functions, usually called modes, allow to express an approximation $u_r$ of $u_h$, such that:

$$u_h(x,t) \approx u_r(x,t) = \sum_{i=1}^{r} \varphi_i(x)\psi_i(t),$$

with $r \leqslant n$, $r$ being called the rank of the approximation. Formally written, modes are defined as:

$$\forall i \in \{1,\dots,r\},\ \exists(\alpha_{1,i},\dots,\alpha_{n,i}) \in \mathbb{R}^n,\ \text{s.t.}\quad \varphi_i(x) = \sum_{j=1}^{n} \alpha_{j,i}\phi_j(x), \quad \forall x \in \Omega.$$

As a result, modes have possibly a larger support on $\Omega$ so that the approached solution can be computed with a reduced number of DOFs. Although fewer in number, each DOF carries more information regarding $u_r$: there should be a duality between the size of the reduced basis and the information each mode allows to express. Additionally, if $u_r$ is a good approximation of $u_h$ for $r \ll n$, that is the rank $r$ of the ROM is a lot smaller than the original number of DOFs $n$, then the ROM is considered to be successful. The main purpose of model reduction

[4] For nodal FE spaces.

[5] Additionally, the choice of basis functions is also crucial to accurately capture the physics being modeled, e.g. avoid issues such as shear locking.

techniques is to provide a method or heuristic to determine the coefficients $(\alpha_{1,i}, \dots, \alpha_{n,i})$. This process is referred to as the offline phase. One can define the reduced counterparts of $\boldsymbol{\phi}_h$ and $\boldsymbol{q}$, respectively, denoted thereafter by $\boldsymbol{\phi}_r$ and $\boldsymbol{q}_r$, such that:

$$u_r(x,t) = \boldsymbol{\phi}_r(x)\boldsymbol{q}_r(t),$$

where:

$$\begin{aligned} \boldsymbol{\phi}_r(x) &= \begin{bmatrix} \varphi_1(x) & \varphi_2(x) & \dots & \varphi_r(x) \end{bmatrix}, \\ \boldsymbol{q}_r(t) &= \begin{bmatrix} \psi_1(t) & \psi_2(t) & \dots & \psi_r(t) \end{bmatrix}^T. \end{aligned}$$

Given the definition of the reduced basis functions, one can define a transformation matrix $\Phi_r \in \mathbb{R}^{n\times r}$ that concatenates the coefficients such that:

$$\boldsymbol{\phi}_r(x) = \boldsymbol{\phi}_h(x)\Phi_r$$

where:

$$\Phi_r = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & & \alpha_{1,r} \\ \vdots & \vdots & \dots & \vdots \\ \alpha_{n,1} & \alpha_{n,2} & & \alpha_{n,r} \end{bmatrix}$$

In the end, one substitutes $u_r(x,t) = \boldsymbol{\phi}_h(x)\Phi_r\boldsymbol{q}_r(t)$ and $u_r^*(x,t) = \boldsymbol{\phi}_h(x)\Phi_r\boldsymbol{q}_r^*(t)$ for $u$ and $u^*$, respectively, in (1.8). And, as previously with (1.11)–(1.13), the problem can be recast in compact form in terms of $\boldsymbol{q}_r$, such that $\boldsymbol{q} = \Phi_r\boldsymbol{q}_r$, as:

$$\begin{aligned} M_r\ddot{\boldsymbol{q}}_r(t) + K_r\boldsymbol{q}_r(t) &= \boldsymbol{f}_r(t), \qquad \forall t \in \mathcal{I}, \\ \boldsymbol{q}_r(0) &= \Phi_r^\dagger\boldsymbol{u}_0, \\ \dot{\boldsymbol{q}}_r(0) &= \Phi_r^\dagger\boldsymbol{v}_0, \end{aligned} \tag{1.14}$$

with $\Phi_r^\dagger$ the left pseudo-inverse of $\Phi_r$. Moreover, $M_r \in \mathbb{R}^{r\times r}$, $K_r \in \mathbb{R}^{r\times r}$, and $\boldsymbol{f}_r : \mathbb{R} \to \mathbb{R}^r$ are given by:

$$\begin{aligned} M_r &= \Phi_r^T M \Phi_r, \\ K_r &= \Phi_r^T K \Phi_r, \\ \boldsymbol{f}_r &= \Phi_r^T \boldsymbol{f}. \end{aligned}$$

The formulation (1.14) may be referred to as the reduced-order model. It is the projection of the full-order model equations onto the reduced subspace spanned by the column vectors of $\Phi_r$. An approximation of the solution of the problem can now be found with $r$ time-continuous DOFs instead of $n$, with $r \ll n$. One can notice that the FE formulation is the particular case where $r = n$ and $\alpha_{i,j} = \delta_{ij}$[6], i.e. $\Phi_r = I_n$.

> **Remark** Formally, the dynamic equilibrium equation is given in terms of the reduced vector of DOFs as:
>
> $$M\Phi_r\ddot{\boldsymbol{q}}_r(t) + K\Phi_r\boldsymbol{q}_r(t) = \boldsymbol{f}(t) + \boldsymbol{r}(t), \tag{1.15}$$
>
> where $\boldsymbol{r}(t)$ is a time-dependent residual. This residual highlights the fact that an error is committed with respect to the full-order equilibrium equation (1.11). At this point, a distinction may be made between the reduced transformation matrix $\Phi_r$ and a projection matrix $\Phi_p$. The projection matrix $\Phi_p$ is usually built considering the Petrov-Galerkin projection [12], such that it is orthogonal to $\boldsymbol{r}(t)$, i.e. $\Phi_p^T\boldsymbol{r}(t) = 0$, and $\Phi_p^T\Phi_r$ is invertible. Thus, the projection of the equilibrium equation reads:
>
> $$\Phi_p^T M\Phi_r\ddot{\boldsymbol{q}}_r(t) + \Phi_p^T K\Phi_r\boldsymbol{q}_r(t) = \Phi_p^T\boldsymbol{f}(t).$$
>
> The first equation of (1.14) is the particular case where the Galerkin projection is considered, i.e. $\Phi_p = \Phi_r$.

The purpose of the next section is to describe the existing ROM techniques and the specific features of the transformation matrix $\Phi_r$ they compute.

## 1.3 Literature review

Several model reduction approaches have been put forth for applications in structural dynamics. Emerging approaches based on Machine Learning are certainly worth being mentioned, e.g. Physics-Informed Neural Networks (PINNs) [13]. However, the present review primarily addresses widely adopted projection-based approaches [8, 14] and the Proper Generalized Decomposition (PGD), which stands out as a distinct approach. This literature review mainly focuses on applications to structural dynamics. For each method, the properties of the reduced basis being built will be emphasized, as well as its advantages and drawbacks.

### 1.3.1 Modal Analysis

The Rayleigh-Ritz method is considered as one of the earliest and foundational model reduction methods in linear dynamics [15]. It typically involves solving an eigenvalue problem and projecting the full-order problem on a basis made of relevant eigenvectors (or eigenmodes). Given the stiffness and mass matrices, one solves the generalized eigenvalue problem $K\boldsymbol{u} =$

[6] $\delta_{ij}$ denotes the Kronecker delta, that is $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise.

$\lambda M\boldsymbol{u}$; such $\lambda \in \mathbb{R}$ and $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{u} \neq 0$, are respectively the eigenvalues and eigenvectors. It undoubtedly remains the most popular technique among engineering analysis tools. If we select $r$ eigenpairs $(\lambda_i, \boldsymbol{u}_i) \in \mathbb{R} \times \mathbb{R}^n$, $i = 1, \ldots, r$, they satisfy the following properties:

$$\begin{aligned} \boldsymbol{u}_i^T K \boldsymbol{u}_j &= k_i \delta_{ij}, \\ \boldsymbol{u}_i^T M \boldsymbol{u}_j &= m_i \delta_{ij}, \end{aligned}$$

where $k_i \in \mathbb{R}_+$ and $m_i \in \mathbb{R}_+^*$, $i = 1, \ldots, r$, can be interpreted as the equivalent stiffness and mass coefficients, respectively, associated with the $i^{\text{th}}$ eigenvector. The eigenvectors are orthogonal to each other with respect to both the mass and stiffness operators. As a result, the mass and stiffness operators are diagonal in the subspace spanned by the eigenvectors. Eigenvectors are unique up to multiplication by a non-zero scalar. Thus, they are usually normalized with respect to the mass operator, such that:

$$\begin{aligned} \boldsymbol{u}_i^T K \boldsymbol{u}_j &= \omega_i^2 \delta_{ij}, \\ \boldsymbol{u}_i^T M \boldsymbol{u}_j &= \delta_{ij}. \end{aligned}$$

with $\omega_i$ the eigen pulsations, such that $\omega_i^2 = k_i/m_i = \lambda_i$, associated with the retained eigenmodes. Eigenvectors are now unique up to the sign. In practice not all the eigenpairs are computed and approaches, such as the shifted block Lanczos algorithm, can be used to compute eigenvectors in the vicinity of an eigenvalue of interest, the latter being used as the shift [16]. Then, Modal Analysis proposes the following choice for $\Phi_r$:

$$\Phi_r = \begin{bmatrix} \boldsymbol{u}_1 & \ldots & \boldsymbol{u}_r \end{bmatrix}.$$

In consequence, the reduced-order system is diagonal and can be written component-wise, such that:

$$\ddot{\psi}_i + \omega_i^2 \psi_i = f_{r,i}, \quad i = 1, \ldots, r.$$

Therefore, not only this method reduces the size of the system to solve, but the components of the reduced vector of DOFs are decoupled.

Yet, eigenvectors only depend on the geometry and material parameters of the structure, hence the reduced model mainly describes the natural response, i.e. when $\boldsymbol{f} = 0$. Consequently, not all eigenvectors are necessarily relevant to obtain the structural response under external loads. Or, conversely, it may introduce a large number of these vectors to describe the mechanical behavior, which is not desirable in reduced-order modeling.

### 1.3.2 Dynamic Substructuring

Later, Component Mode Synthesis [17] techniques were developed. Among them, the Craig and Bampton method [18] has been one of the most prolific. It combines features from modal decomposition and static condensation to construct a reduced-order model that accurately represents the dynamic behavior of substructures or assemblies. Nodes are partitioned into two domains, such that the dynamic equilibrium equation reads:

$$\begin{bmatrix} M_{ii} & M_{ib} \\ M_{bi} & M_{bb} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_i(t) \\ \ddot{\boldsymbol{q}}_b(t) \end{bmatrix} + \begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}_i(t) \\ \boldsymbol{q}_b(t) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_i(t) \\ \boldsymbol{f}_b(t) \end{bmatrix},$$

where vectors subscripted by $i$ and $b$ denote **i**nterior and interface **b**oundary DOFs, respectively; operators subscripted by $ii$ and $bb$ denote material properties associated to interior and interface DOFs, while subscripts $ib$ and $bi$ stand for the coupling between DOFs of the two partitions. The choice made in the Craig-Bampton methodology is given by:

$$\Phi_r = \begin{bmatrix} \Psi_b & \Psi_i \end{bmatrix}, \quad \text{with :} \quad \begin{aligned} \Psi_b &= \begin{bmatrix} -K_{ii}^{-1} K_{ib} \\ I_{bb} \end{bmatrix}, \\ \Psi_i &= \begin{bmatrix} \Phi \\ 0 \end{bmatrix}, \end{aligned}$$

where $I_{bb}$ is the identity matrix whose dimension is the number of interface DOFs and $\Phi$ is the matrix that concatenates a set of eigenvectors of the interior structures. Assuming the interface boundaries to be clamped, the eigenvectors are the solutions to the generalized eigenvalue problem $K_{ii}\boldsymbol{u} = \lambda M_{ii}\boldsymbol{u}$. $\Psi_b$ and $\Psi_i$ are commonly referred to as the constraint modes and the fixed-interface normal modes, respectively. The retained eigenvectors define the frequency bandwidth on which such a ROM is valid [19]. This particular mapping $\Phi_r$ leads to a block diagonal reduced stiffness matrix:

$$K_r = \Phi_r^T K \Phi_r = \begin{bmatrix} K_{bb} - K_{bi} K_{ii}^{-1} K_{ib} & 0 \\ 0 & \Omega_{ii}^2 \end{bmatrix}$$

where $\Omega_{ii}$ is the diagonal matrix of the retained eigen pulsations of the interior partition. This structure highlights the orthogonality of the constraint modes with the fixed-interface normal modes with respect to the stiffness operator. As a result, $K_r$ is mostly sparse. This property is leveraged to efficiently compute large numbers of eigenpairs [20] with the Automated Multilevel Substructuring Method (AMLS).

Nonetheless, it can be observed that the dimension of the reduced model is driven by the number of DOFs on the interface, rather than by the actual behavior of the structure [21].

### 1.3.3 Proper Orthogonal Decomposition

An alternative approach that circumvents the latter shortcoming is the POD. This data-driven approach builds a reduced model generally obtained by the Singular Value Decomposition (SVD) performed on snapshots describing the evolution of the states of the system. It can be viewed as an *a posteriori* approach, in the sense that it takes the state of the full-order model at different time-steps[7] as an input, the so-called snapshots, in order to extract the dominant modes in the data. Given a snapshot matrix $S \in \mathbb{R}^{n \times m}$, such that:

$$S = \begin{bmatrix} \boldsymbol{q}(t_1) & \dots & \boldsymbol{q}(t_m) \end{bmatrix},$$

the following optimization problem has to be solved [22]:

$$\begin{aligned} \underset{\Phi_r \in \mathbb{R}^{n \times r}}{\text{minimize}} \quad & \|S - \Phi_r \Phi_r^T S\|, \\ \text{subject to} \quad & \Phi_r^T \Phi_r = I_r, \end{aligned}$$

where $\|\cdot\|$ is the Frobenius norm. The Schmidt-Mirsky-Eckart-Young theorem [23] states that the truncated SVD of $S$ gives the solution to the above minimization problem. The SVD[8] computes a factorization of $S$ as:

$$S = U \Sigma V^T,$$

where, with $r \leqslant \min(n, m)$, $U \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{m \times r}$ verify the following properties:

- $U^T U = I_r$ and $V^T V = I_r$, i.e. $U$ and $V$ are semi-orthogonal;
- $\Sigma$ is diagonal with strictly positive numbers.

The reduced basis is then defined with the column vectors of $U$:

$$\Phi_r = U.$$

In the case of dynamics, the column vectors of $V$ can be used as a reduced basis for time functions and are referred to as temporal modes.

[7] In the case where the time variable is considered to be the parameter (time-domain POD).

[8] The real, compact SVD variant is described here.

The performance of the POD has been demonstrated for both linear and nonlinear dynamics subjected to transient loads [24–26], and most notably, within a goal-oriented framework for elastodynamics, in the cases of either space-time approximation [27] or parametrized problems [28]. One well-known shortcoming of the Proper Orthogonal Decomposition (POD) is that it requires snapshots computed in an offline stage by an expensive high-fidelity solver, e.g. the Finite Element Method. In [29, 30], the snapshots are evaluated at a given time-step only if the estimated error exceeds a given threshold and reduced bases are subsequently enriched in an adaptive manner. Such an approach allows one to mitigate the cost in evaluating the full-order model. Similarly, the so-called Reduced Basis (RB) technique provides a snapshot selection strategy designed to minimize the error committed by the reduced model [31, 32].

The reduced mapping $\Phi_r$ computed therein is chosen semi-orthogonal by convention with respect to the Euclidean norm. Depending on the mechanical properties, e.g. heterogeneity, this norm may not allow one to capture the physics being modeled. In that regard, energy norms may be preferred instead [25, 33]. Furthermore, the authors in [8] pointed out that the offline phase is somewhat suboptimal when the goal is to use the ROM for optimization. Indeed, the offline phase involves an exploration of the parametric space, but it is not known a priori whether the explored regions will subsequently be exploited throughout the online phase (optimization); or worse, the optimization process could lead to assess the response of the system in some regions where the offline phase did not gather enough information about the system's behavior. This remark especially applies to methods like the POD, where the exploration is contingent to costly, full-order solutions [34].

### 1.3.4 Proper Symplectic Decomposition

The PSD is specifically concerned with the preservation of the symplectic structure of the full-order model. This structure arises in the context of Hamiltonian systems, which will be introduced thereafter.

The structural dynamics problem that we are interested in can be described in terms of the Hamiltonian functional. The Hamiltonian formalism consists in modeling the motion of the system along a trajectory in the phase space by introducing the generalized coordinates $\boldsymbol{q}$ and their generalized (or conjugate) momenta $\boldsymbol{p}$ as independent variables. For the problem at hand, the Hamiltonian functional $h$ reads:

$$h(\boldsymbol{q}, \boldsymbol{p}, t) = \frac{1}{2}\boldsymbol{q}^T K \boldsymbol{q} + \frac{1}{2}\boldsymbol{p}^T M^{-1} \boldsymbol{p} - \boldsymbol{q}^T \boldsymbol{f}. \tag{1.16}$$

The equilibrium equation (1.11), restated in terms of the Hamiltonian, is equivalent to the so-called Hamilton's equations [35, 36]:

$$\begin{aligned}\dot{\boldsymbol{q}} &= \quad \nabla_p\, h,\\ \dot{\boldsymbol{p}} &= -\nabla_q\, h.\end{aligned}$$

Let us introduce the canonical coordinates $\boldsymbol{z}$ that vertically concatenates $\boldsymbol{q}$ and $\boldsymbol{p}$ such that:

$$\boldsymbol{z} = \begin{bmatrix}\boldsymbol{q}\\ \boldsymbol{p}\end{bmatrix}.$$

The gradient of the Hamiltonian (1.16) then reads:

$$\nabla_z\, h = \begin{bmatrix}\nabla_q\, h\\ \nabla_p\, h\end{bmatrix} = \begin{bmatrix}K\boldsymbol{q} - \boldsymbol{f}\\ M^{-1}\boldsymbol{p}\end{bmatrix}.$$

In the symplectic framework, the dynamics of the structure is modeled by the trajectory in the symplectic vector space $(\mathbb{R}^{2n}, \omega)$ of dimension $2n$ for linear systems, where $\omega$ is the so-called symplectic form defined as:

$$\forall \boldsymbol{z} = \begin{bmatrix}\boldsymbol{q}\\ \boldsymbol{p}\end{bmatrix} \in \mathbb{R}^{2n}, \quad \forall \boldsymbol{z'} = \begin{bmatrix}\boldsymbol{q}'\\ \boldsymbol{p}'\end{bmatrix} \in \mathbb{R}^{2n}, \quad \omega(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{q}^T\boldsymbol{p}' - \boldsymbol{q}'^T\boldsymbol{p} = \boldsymbol{z}^T J_{2n}\boldsymbol{z}',$$

with $J_{2n}$ the skew-symmetric operator such that:

$$J_{2n} = \begin{bmatrix}0 & I_n\\ -I_n & 0\end{bmatrix},$$

and $J_{2n}^2 = -I_{2n}$. It is then possible to recast (1.11) as:

$$\dot{\boldsymbol{z}} = \nabla^{\omega} h,$$

where $\nabla^{\omega} = J_{2n}\nabla_z$ is defined as the symplectic gradient. The Hamiltonian can be written as a sum of a quadratic form on $\mathbb{R}^{2n}$ and the external energy term:

$$h(\boldsymbol{z}, t) = \frac{1}{2}\boldsymbol{z}^T H\boldsymbol{z} - \boldsymbol{z}^T\boldsymbol{f}_z,$$

with $H$ the Hessian operator of $h$ and $\boldsymbol{f}_z$ such that:

$$H = \begin{bmatrix}K & 0\\ 0 & M^{-1}\end{bmatrix}, \qquad \boldsymbol{f}_z = \begin{bmatrix}\boldsymbol{f}\\ 0\end{bmatrix}.$$

We now introduce the notion of symplectic mapping. A symplectic mapping is a linear transformation that preserves the symplectic form $\omega$, i.e.:

$$A \in \mathbb{R}^{2n\times 2n} \text{ is symplectic if } \quad \omega(A\boldsymbol{z}, A\boldsymbol{z}') = \omega(\boldsymbol{z}, \boldsymbol{z}'), \quad \forall(\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}.$$

As a consequence, such a mapping $A$ verifies:

$$A^T J_{2n} A = J_{2n}.$$

The notion can actually be generalized to rectangular matrices with the symplectic Stiefel manifold, denoted $S_p(2r, 2n)$, such that:

$$S_p(2r, 2n) = \left\{ A \in \mathbb{R}^{2n\times 2r} : \; A^T J_{2n} A = J_{2r} \right\}. \tag{1.17}$$

Let $(\mathbb{R}^{2r}, \gamma)$, with $r \ll n$, be a symplectic vector space, $A \in S_p(2r, 2n)$ a symplectic mapping, and $\boldsymbol{y} \in \mathbb{R}^{2r}$ such that $\boldsymbol{z} = A\boldsymbol{y}$. One can define a Hamiltonian for the reduced canonical coordinates $\boldsymbol{y}$:

$$g(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^T G\boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{f}_y,$$

with $G$ its Hessian operator and $\boldsymbol{f}_y$ the projection of the external loads onto the symplectic subspace (in the case $r \leqslant n$), such that:

$$G = A^T H A, \quad \text{and} \quad \boldsymbol{f}_y = A^T \boldsymbol{f}_z.$$

The system to be solved is governed by the reduced Hamiltonian $g$, whose Hessian $G$ can be interpreted as the reduced counterpart of the Hessian operator $H$. The preservation of the symplectic structure implies that $\boldsymbol{y}$ is governed by Hamilton's canonical equations, expressed hereinafter in terms of $\gamma$ (symplectic form on $\mathbb{R}^{2r}$) and $g$ such that:

$$\dot{\boldsymbol{y}} = \nabla^\gamma g,$$

with $\nabla^\gamma = J_{2r}\nabla_y$.

In Hamiltonian mechanics, symplecticity-preserving methods are especially relevant because they exhibit good accuracy. Symplectic integrators are particularly robust to compute long-time evolution of Hamiltonian systems [37–39]. In addition, symplectic methods were found proficient for the treatment of elastodynamics problems that involve large rotations and small strains [39].

SVD-based methods, such as the cotangent lift and complex SVD techniques, are prominent in generating symplectic bases [40, 41]. The cotangent lift method involves performing the SVD on covariance matrices for generalized coordinates and conjugate momenta. It effectively preserves the symplectic structure of the original system. The complex SVD method, which represents the Hamiltonian system in a complex form and applies the SVD to this representation, is useful for systems with oscillatory behavior or complex-valued states but may introduce additional computational complexity. In the case of periodic solutions, a recent work shows that the POD of canonizable Hamiltonian systems naturally leads to a symplectic basis [42]. Aside, the greedy approaches were applied to construct a symplectic basis iteratively, selecting basis vectors that maximize the energy content or minimize an error with respect to a chosen criterion [40, 41]. This method is adaptable and efficient in capturing significant modes of the Hamiltonian system, but it requires careful selection of the criterion and stopping condition to ensure optimal performance. Like most greedy approaches, the basis vectors must be orthonormalized. Yet, the classic orthonormalization algorithms do not preserve the symplectic structure [43]. Therefore, the symplectic Gram-Schmidt method [44] is considered instead. On another note, Hamiltonian operator inference involves learning a reduced-order model directly from data by inferring the reduced Hamiltonian operator [45]. The strategies described above require to project the full-order model onto the symplectic subspace, as shown in Eq. (1.14). Conversely, Hamiltonian operator inference solely relies on data and removes the need of a complete knowledge of the full-order model operators. In that regard, this technique is said to be nonintrusive. Another data-based PSD draws upon modal analysis. The data is interpreted in terms of the symplectic formalism to define two intertwined eigenvalue problems whose solutions are the PSD modes [46]. Like other *a posteriori* methods, these approaches may become computationally intensive with the full-order data generation for large-scale systems.

### 1.3.5 Proper Generalized Decomposition

The PGD [47,48] constructs a reduced basis on-the-fly, eliminating the need for prior knowledge of the solution to the problem. In that respect, the PGD method is used as an *a priori* approach and can be assimilated as a solver: one simultaneously solves the problem and constructs a reduced approximation subspace. The PGD strategy is akin to the method of separation of variables, in the sense that one assumes that the solutions to differential equations are separable with respect to the independent variables and/or the model parameters. Without accounting for model parameters, one would assume that the solutions to the elastodynamics equation are space-time separable. In other words, that the fields can be approximated by a sum of products of functions in space by functions in time, such that:

$$u_h(x,t) \approx u_r(x,t) = \sum_{i=1}^{r} \varphi_i(x)\psi_i(t),$$

where $\varphi_i$ and $\psi_i$, $i = 1, \dots, r$, are referred to as spatial and temporal modes, respectively. This representation of the approximate solution is actually common to all the methods previously mentioned. Yet, the PGD differs from those methods as the assumed separated representation for $u_r$ is injected into the weak formulation: the $\varphi_i$ and $\psi_i$ become the new unknowns of the problem. The Galerkin-based definition of the PGD is detailed thereafter.

Let us consider a problem under the following residual form:

$$\text{Find } u \in V_{\mathcal{R}}, \text{ such that:} \quad \mathcal{R}(u,u^*) = \mathcal{F}(u^*) - \mathcal{B}(u,u^*) = 0, \quad \forall u^* \in V_{\mathcal{R}}^*,$$

where $\mathcal{F}$ is linear, $\mathcal{B}$ is bilinear, and $V_{\mathcal{R}}$ and $V_{\mathcal{R}}^*$ are the spaces of trial and test functions, respectively. They depend on the residual operator $\mathcal{R}$. The PGD reduced model is iteratively computed, usually with one enrichment at a time. Let us begin with the first enrichment $u_1$, sought for as:

$$u_1(x,t) = \varphi(x)\psi(t),$$

and test functions are considered in the form:

$$u^*(x,t) = \varphi^*(x)\psi(t) + \varphi(x)\psi^*(t).$$

The substitution of $u$ by $u_1$ and $u^*$ by the expression above in the residual operator reads:

Find $(\varphi,\psi) \in \mathcal{S}_{\mathcal{R}} \times \mathcal{T}_{\mathcal{R}}$, such that:

$$\begin{aligned} &\mathcal{R}(\varphi\psi, \varphi^*\psi + \varphi\psi^*) = 0, \\ \Leftrightarrow\ &\mathcal{R}(\varphi\psi, \varphi^*\psi) + \mathcal{R}(\varphi\psi, \varphi\psi^*) = 0,\ \forall(\varphi^*,\psi^*) \in \mathcal{S}_{\mathcal{R}}^* \times \mathcal{T}_{\mathcal{R}}^*, \end{aligned} \tag{1.18}$$

where $\mathcal{S}_{\mathcal{R}}$, $\mathcal{T}_{\mathcal{R}}$, $\mathcal{S}_{\mathcal{R}}^*$, and $\mathcal{T}_{\mathcal{R}}^*$ are appropriate function spaces. As a result, the new formulation of the problem is no longer linear in the unknowns. Such a problem could be solved by means of a nonlinear solver like the Newton-Raphson's method. However, the fixed-point iteration is generally preferred. It consists in solving the problem in two steps:

1. Solve (1.18) for $\varphi$ with $\psi$ known. This step is referred to as the spatial problem and reads:

$$\mathcal{R}(\varphi\psi, \varphi^*\psi) = 0, \quad \forall \varphi^* \in \mathcal{S}_{\mathcal{R}}^*. \tag{1.19}$$

2. Solve (1.18) for $\psi$ with $\varphi$ known. This step is referred to as the temporal problem and reads:

$$\mathcal{R}(\varphi\psi, \varphi\psi^*) = 0, \quad \forall \psi^* \in \mathcal{T}_{\mathcal{R}}^*. \tag{1.20}$$

Steps 1 and 2 are repeated until a convergence criterion is fulfilled. We can see here the advantage of the fixed-point scheme, as one parameter dependency is treated at a time. The theoretical complexity of PGD solvers also decreases compared to conventional solvers. Indeed, according to [48], one observes that the complexity of conventional solvers scales exponentially with the dimensions of the problem, while that of PGD solvers scales linearly. For example, considering $n$ spatial DOFs and $n_t$ temporal DOFs, the global dimension of the problem is $n \times n_t$. The combination of the PGD strategy with a fixed-point iteration leads to two subproblems (1.19) and (1.20), whose complexities are solely subject to the dimensions $n$ and $n_t$, respectively: it reduces high-dimension problems into subproblems of lower dimensions. Conversely, the Newton-Raphson algorithm tackles the problem in a coupled fashion and does not allow for such a complexity reduction.

Once the fixed-point iteration has converged, the product of modes $\varphi$ and $\psi$ is the first enrichment of the PGD approximation and will be written $\varphi_1\psi_1$. Then, subsequent enrichment terms can be computed. Let $m \in \mathbb{N}$, $m > 1$, be such that the $m^{\text{th}}$ enrichment is computed considering that the first $m-1$ pairs of modes are known. The $m^{\text{th}}$ pair of modes is sought for such that $u_m$, the decomposition of rank $m$, writes:

$$u_m(x,t) = \underbrace{\sum_{i=1}^{m-1} \varphi_i(x)\psi_i(t)}_{\text{known decomposition}} + \underbrace{\varphi(x)\psi(t)}_{\text{new enrichment}},$$

and the following fixed-point iterations are subsequently solved for as many enrichment terms as needed:

$$\mathcal{R}(\varphi\psi, \varphi^*\psi) = -\sum_{i=1}^{m-1} \mathcal{R}\left(\varphi_i\psi_i, \varphi^*\psi\right), \quad \forall \varphi^* \in \mathcal{S}_{\mathcal{R}}^*,$$

$$\mathcal{R}(\varphi\psi, \varphi\psi^*) = -\sum_{i=1}^{m-1} \mathcal{R}\left(\varphi_i\psi_i, \varphi\psi^*\right), \quad \forall \psi^* \in \mathcal{T}_{\mathcal{R}}^*.$$

In other words, for a given decomposition, the computation of a new enrichment is based on a residual that assesses the error committed by the current ROM. In that regard, enrichment terms are interpreted as corrections: new modes are meant to retrieve missing information or figuratively “fill the blanks”. The rank $r$ of the decomposition is defined as the final number of enrichment $m$. As an example, for a one-dimensional medium $\Omega$ with density $\rho$ and Young’s modulus $E$, the insertion of the first enrichment into the continuous dynamic equilibrium equation (1.1) reads (with $\varphi'' = d^2\varphi/dx^2$):

$$\rho\varphi_1(x)\ddot{\psi}_1(t) - E\varphi_1''(x)\psi_1(t) = f(x,t) + r_1(x,t), \qquad \forall (x,t) \in \Omega \times \mathcal{I},$$

where $r_1 : \Omega \times \mathcal{I} \to \mathbb{R}$ is a space-time strong residual that characterizes the error committed by the ROM with respect to the full-order equilibrium equation. For a given rank $m$, $1 \leqslant m \leqslant r$, the expression above writes:

$$\rho\sum_{i=1}^{m}\varphi_i(x)\ddot{\psi}_i(t) - E\sum_{i=1}^{m}\varphi_i''(x)\psi_i(t) = f(x,t) + r_m(x,t), \qquad \forall (x,t) \in \Omega \times \mathcal{I},$$

with $r_m$ the $m^{\text{th}}$ strong residual. Thus, further enrichment terms are computed such that $\|r_i\| \to 0$ as $i$ increases, in the sense of a chosen norm $\|\cdot\|$.

Ultimately, the PGD also allows one for computing separated representations for fields that depends on parameters other than space and time [14], e.g. with $\wp \in \mathbb{N}$ extra-parameters, such that:

$$u_h(x,t,p_1,\ldots,p_\wp) \approx u_r(x,t,p_1,\ldots,p_\wp) = \sum_{i=1}^{r}\varphi_i(x)\psi_i(t)\prod_{j=1}^{\wp}\mu_i^j(p_j),$$

and the fixed-point iteration described above can be generalized, with one subproblem per parameter to be solved for the modes $\mu_i^j$.

Several efforts have been deployed in the last decade towards building separated approximations of solutions governed by second-order hyperbolic equations. However, the performance of the PGD approach using a space-time separation for transient structural dynamics has often been considered unsatisfactory [49, 50]. One reason is that the fixed-point algorithm employed by Galerkin-based PGD solvers tends to exhibit poor convergence, if it converges at all [49]. In fact, it is open to question whether the PGD framework using space-time separability is suitable for solving second-order hyperbolic problems. Of particular interest is the fact that low rank decomposition of high-order tensors, i.e. whenever the order is higher than unity, inevitably leads to ill-posed formulations [51]. This is mainly due to non-uniqueness of such separated representations [52]. This issue has been overcome with the implementation of a greedy rank-one strategy, where one mode per parameter is computed, combined with an update of formerly computed temporal modes [53]. The PGD does not inherently compute spatial modes with a specific structure; it is up to the user to enforce any desired structure for the spatial modes. Their normalization and/or orthogonalization by means of any Gram-Schmidt process usually improves the robustness of the algorithm [54]. The authors in [49, 55] have also proposed an alternative to the Galerkin-based PGD, namely the minimal residual PGD, with an emphasis on optimal norms for the PGD in elastodynamics. The minimal residual PGD consistently converges, a proof of which is given in [56]. Different approaches have also been proposed in order to circumvent the difficulty one encounters when using a separated representation in space and time. For instance, an approach that assumes a good space separation, was presented in [57]. It consists in estimating, in an adaptive manner, the number of spatial modes at each time step. Alternatively, space-frequency separated representations were developed for a fairly wide range of applications: 2D acoustics [58, 59], nonlinear soil mechanics [60], linear and nonlinear structural dynamics [61, 62] and transient electronics [63]. The space-frequency separation was shown to be particularly efficient if additional parameters (material, geometric, etc.) were accounted for. In this respect, the advantage of PGD solvers seems clear when geometric or material parameter separation is at stake, offering a considerable reduction of the computational complexity [48, 50]. However, the space-frequency formulation does not necessarily provide direct insights into the transient behavior of the system. While it can determine its response at specific frequencies, it may fail to accurately capture time-dependent loads or dynamical events. Recent works have introduced a variant of the PGD, called sparse PGD (sPGD). The latter is at the core of a multi-parametric approach in the context of digital twins [64]. The sPGD method was used in an *a posteriori* setting with scarce input data and in combination with a Harmonic-Modal Hybrid (HMH) solver and the Discrete Empirical Interpolation Method (DEIM). Furthermore, the PGD framework has also been extended to perform basic operations (multiplications, divisions) as well as more complex operations, such as solving linear systems of algebraic equations, leveraging the principle of variable separation. The authors in [65] have thus created a versatile toolbox for PGD algebraic operators, which has been used in a non-intrusive manner to solve parametric eigenproblems arising, for instance, in automotive applications [66, 67].

### 1.3.6 Viscous damping in ROM for structural dynamics

In the case of viscous damping, the equilibrium equation of the full-order model becomes:

$$M\ddot{\boldsymbol{q}}(t) + C\dot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I},$$

where $C$ is the global damping matrix. Similar to the stiffness and mass operators, $C$ is assembled based on a tensor that represents the viscous local properties of the medium $\Omega$. However, characterizing the local viscous properties remains challenging [68]. Therefore, the assumption of a proportional damping is often employed to model the structure's damping, that is the damping matrix $C$ is written as $C = \alpha K + \beta M$ [68–70], with $\alpha$, $\beta \in \mathbb{R}_+$. This damping formulation is also called Rayleigh damping and $\alpha$ and $\beta$ are referred to as the

Rayleigh damping parameters. These coefficients do not have a clear physical interpretation regarding the materials [69, 70]. Their estimation remains nevertheless an active research topic in both the experimental and numerical communities [71, 72], as the control of damping phenomena is a critical issue in the construction, automotive, or aerospace industries.

This hypothesis is particularly convenient in Modal Decomposition since the damping operator is also diagonal in the eigenspace. The reduced-order system can still be written component-wise, such that:

$$\ddot{\psi}_i + 2\xi_i\omega_i\dot{\psi}_i + \omega_i^2\psi_i = f_{r,i}, \quad i = 1, \dots, r,$$

with $\xi_i$ being the modal damping coefficients, given by:

$$\xi_i = \frac{1}{2}\left(\alpha\omega_i + \frac{\beta}{\omega_i}\right).$$

Thus, the same eigenvalue problem as for the undamped equation, i.e. $\boldsymbol{K}\boldsymbol{u} = \lambda\boldsymbol{M}\boldsymbol{u}$, allows one to estimate the damped behavior of a structure for any values of $\alpha$ and $\beta$. One can see that the modal damping term $\xi_i$ is proportional to the modal pulsation $\omega_i$ with respect to $\alpha$; conversely, it is inversely proportional to the modal pulsation with respect to $\beta$. As a result, proportional damping may produce underdamped or overdamped behaviors in certain frequency ranges [68] (although this may be convenient in some cases). In practice, the goal is to optimize parameters $\alpha$ and $\beta$ around a pulsation of interest. Similarly to Modal Decomposition, several methods based on Krylov subspaces have been developed to build proportionally damped reduced models [73–75]. However, both Modal Decomposition and Krylov-based methods operate in two steps:

1. Build a ROM for the undamped equation;
2. Project the damped equation onto the reduced basis for given values of $\alpha$ and $\beta$.

Hence, the reduced model is actually not parametrized with respect to the damping parameters.

## 1.4 Thesis objective and contributions

None of the methods presented above stands out as an indisputable candidate for structural dynamics. Some oppositions can be highlighted: the POD is robust and weakly intrusive but full-order solves are costly; on the other hand, the PGD may suffer from convergence issues, but its framework is versatile and removes the burden of full-order solves. On another note, the

PSD specifically tackles the preservation of the symplectic structure of the original problem but, like the POD, suffers from costly evaluations when applied to large-scale full-order models.

> **Main objective** The main objective of the thesis is to propose novel implementations that offer:
>
> 1. Robustness, computational efficiency, and preservation of the symplectic structure of the problem (when applicable);
>
> 2. The parametrization of the ROM with respect to the Rayleigh damping parameters.

In the light of this literature review and the scope of this work, the framework provided by the PGD seems to be the most appealing. The method is versatile regarding:

- The strategy to obtain the solutions to the subproblems (linear solver, time integration, etc.);
- The constraints or properties imposed on the modes (e.g., normalization and/or orthogonalization and the choice of a metric);
- The parameters to be accounted for.

One should take advantage of this flexible framework and tune it towards the main objective of the thesis. To this end, three specific objectives **SO1**, **SO2**, and **SO3** were set as follows.

> **SO1: Investigate the convergence issue of the Galerkin-based PGD formulation and develop a robust implementation**
> Drawing upon the work presented in [49,50,55], the lack of convergence of the space-time Galerkin-based PGD was investigated. An algorithm was implemented following a greedy rank-one strategy with an orthonormalization of the spatial modes [54] and an update of the temporal mode after each enrichment [53]. The novelty of this work lied in the development of weak formulations of the PGD problems based on the Lagrangian and Hamiltonian Mechanics, the objective being to devise numerical methods that are numerically stable and energy conservative. The solver based on the Hamiltonian formulation was showed to offer better stability and energy conservation properties than that based on the Lagrangian formulation.

This objective led to the following scientific communications:

- Peer-reviewed article

  **C. Vella**, S. Prudhomme, "PGD reduced-order modeling for structural dynamics applications", *Computer Methods in Applied Mechanics and Engineering*, Vol. 402, p. 115736, 2022, https://doi.org/10.1016/j.cma.2022.115736.

- Conference presentation (*: speaker)

  **C. Vella***, S. Prudhomme, "On the stability of PGD reduced-order models for structural dynamics applications". Oral presentation at the 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS) in Oslo, Norway, on June 5–9, 2022.

> **SO2: Extend the base implementation towards higher scales and the preservation of the symplectic structure with respect to the time variable**
> The second specific objective primarily focuses on enhancing the computational efficiency of the previously introduced PGD solver based on the Hamiltonian formalism. The novelty of this development lied in the implementation of a solver that is halfway between Modal Decomposition and the conventional PGD framework, to specifically reduce the cost of the spatial solver. The original idea consists in preprocessing the eigenpairs approximations of the operators, namely the Ritz pairs, that provide a subspace in which the problem in space remains diagonal throughout the fixed-point iterations. All computations are then carried out in the subspace spanned by the Ritz vectors, hence drastically decreasing the computational burden. Additional procedures such as Aitken's delta-squared process and mode-orthogonalization are incorporated to ensure convergence and stability of the algorithm. Numerical results regarding the ROM accuracy, time complexity, and scalability were provided to demonstrate the performance of the new solver when applied to dynamic simulation of a three-dimensional structure.

This objective led to the following scientific communications:

- Peer-reviewed article

  **C. Vella**, P. Gosselet, S. Prudhomme, "An efficient PGD solver for structural dynamics applications", *Advanced Modeling and Simulation in Engineering Sciences*, Vol. 11, 2024, https://doi.org/10.1186/s40323-024-00269-z.

- Conference presentations (*: speaker)

  **C. Vella***, S. Prudhomme, “Symplectic formulation of PGD reduced-order models for structural dynamics applications”. Poster presentation at the Model Reduction and Surrogate Modeling (MORE) conference in Berlin, Germany, on September 19–23, 2022.

  **C. Vella**, P. Gosselet, S. Prudhomme*, “On an efficient PGD solver for structural dynamics applications”. Oral presentation at the 6th International Workshop on Model Reduction Techniques (MORTech 2023) in Paris-Saclay, France, on November 22–24, 2023.

> **SO3: Develop a parametrized PGD reduced model with respect to Rayleigh damping coefficients and enhance the efficiency and accuracy of the spatial solver**
> The third and final objective focuses on the development of a PGD reduced model that is parametrized by Rayleigh damping coefficients. The proposed approach incorporates damping modes to construct a damped surrogate model effectively. A novel method has been introduced to solve the problem in space: during the offline phase, the spatial problem is initially projected onto the subspace spanned by the Ritz vectors of the system to provide an efficient prediction of the spatial modes. The prediction is then refined using a MinRes iterative solver. This two-step process, akin to a prediction-correction method, reduces the computational cost of a full-order solution while improving the accuracy of the reduced model. The resulting PGD surrogate has been subsequently employed within a Particle Swarm Optimization algorithm to determine optimal damping coefficients based on a given snapshot. Numerical experiments has demonstrated the effectiveness of the proposed approach.

This objective led to the following scientific communications:

- Peer-reviewed article

  **C. Vella**, S. Prudhomme, “PGD surrogate modeling with application to the identification of Rayleigh damping parameters”, submitted to *Computers & Structures.*

- Conference presentation (*: speaker)

  **C. Vella***, S. Prudhomme, “On an efficient PGD solver for damped elastodynamics optimization”. Oral presentation at the 16th World Congress on Computational Mechanics (WCCM) and 4th Pan American Congress on Computational Mechanics (PANACM) in Vancouver, Canada, on July 21–26, 2024.

# CHAPTER 2 ON A GALERKIN-BASED PROPER GENERALIZED DECOMPOSITION FOR THE 1D WAVE EQUATION: DEVELOPMENT OF A ROBUST DISPLACEMENT-MOMENTUM APPROACH

In this chapter, novel space-time separated representations are developed for the wave equation with the objective of devising numerical methods that are stable and energy conservative. Particular attention is paid to the derivation of weak formulations, at the continuous level, following the application of the Hamilton's Weak Principle [76]. The first formulation is based on the Lagrangian description of Mechanics [77], where only the displacement field is considered as an unknown of the problem. The second one is based on the Hamiltonian theory [35, 36], in which both the displacement field (generalized coordinates) and the conjugate field (generalized momenta) are treated as unknown fields. We thus derive two weak formulations that allow us to implement the Galerkin-based version of the PGD, which will be referred to as L-PGD and H-PGD, respectively. In particular, the Hamiltonian approach naturally leads to a mixed weak formulation that allows one to introduce two separated representations, one for the displacement field and the other for the conjugate field, in a manner similar to the MF-PGD [49], but using the Galerkin-based PGD.

Regarding the discretization in time, several integration schemes have been adapted to linear elastodynamics [15, 78, 79]. Only stable, energy conservative discretization schemes are considered here, namely the Crank-Nicolson method [80–82] (also known as the implicit trapezoidal rule), and the Newmark method [83] with $\gamma = 1/2$ and $\beta = 1/4$. Moreover, we apply two post-processing procedures that aim at improving the convergence of the Galerkin-based version of the PGD [53, 54], namely 1) the orthogonalization of the spatial modes via a modified Gram-Schmidt algorithm, and 2) the update procedure of the temporal modes. These procedures are applied to both the L-PGD and H-PGD. We will show in the numerical examples that the H-PGD solver has a better numerical stability and produces solutions with better energy conservation than the L-PGD, and this for all implemented test cases. One reason is that the orthonormalization and update procedures for the H-PGD truly work in synergy. Moreover, we propose an adaptive fixed-point algorithm for the H-PGD that independently controls, and thus accelerates, the convergence of the fields. Finally, we will show through a numerical example that the methodology can be extended in a straightforward manner to the case of the wave equation involving a linear damping term.

The chapter is organized as follows: in Section 2.1, we describe the model problem and provide an analytical solution by the method of separation of variables for a specific set of initial and boundary conditions. In Section 2.2, we present the weak formulations of the problem based on the Lagrangian and Hamiltonian formalisms and derive discrete counterparts using the Finite Element method in space and numerical integration schemes in time. The L-PGD and H-PGD approaches are described in Section 2.3 along with the orthogonalization and updating procedures as well as the fixed-point algorithms. Numerical experiments are presented in Section 2.4 to illustrate the performance of the proposed approaches. We finally provide some concluding remarks in Section 2.5.

## 2.1 Model problem

### 2.1.1 Strong formulation

The model problem we shall consider consists of a 1D bar in traction or compression under the assumption of infinitesimal deformation. The bar has density $\rho$, Young's modulus $E$, length $\ell$, and cross-sectional area $A$. We will assume that $E$ and $A$ are constant but that $\rho$ could possibly vary in space. Let $\Omega = (0, \ell)$ be the open interval in $\mathbb{R}$ occupied by the bar and let $\mathcal{I} = (0, T)$ denote the time interval. The displacement $u = u(x, t)$ is governed by the 1D wave equation:

$$\rho A \frac{\partial^2 u}{\partial t^2} - EA \frac{\partial^2 u}{\partial x^2} = f, \qquad \forall (x, t) \in \Omega \times \mathcal{I}, \tag{2.1}$$

and subjected to the initial conditions:

$$u(x, 0) = u_0(x), \qquad \forall x \in \Omega, \tag{2.2}$$

$$\frac{\partial u}{\partial t}(x, 0) = v_0(x), \qquad \forall x \in \Omega, \tag{2.3}$$

as well as to the boundary conditions:

$$u(0, t) = 0, \qquad \forall t \in \mathcal{I}, \tag{2.4}$$

$$EA \frac{\partial u}{\partial x}(\ell, t) = g(t), \qquad \forall t \in \mathcal{I}, \tag{2.5}$$

where the functions $f = f(x, t)$, $u_0 = u_0(x)$, $v_0 = v_0(x)$, and $g = g(t)$ are supposed sufficiently regular to yield a well-posed problem. In the following, we will denote the time derivatives by $\dot{u} = \partial u / \partial t$ and $\ddot{u} = \partial^2 u / \partial t^2$ and the space derivatives by $u' = \partial u / \partial x$ and $u'' = \partial^2 u / \partial x^2$. Moreover, we introduce the wave speed as $c = \sqrt{E/\rho}$.

### 2.1.2 Analytical solution

In the case that the speed $c$ is chosen constant, it is well known that the general solution to the homogeneous wave equation (2.1) in an infinite domain, i.e. with $f(x,t) = 0$ and $\Omega = \mathbb{R}$, can be recast, using the d'Alembert formula, as $u(x,t) = \varphi(x+ct) + \phi(x-ct)$, where $\varphi$ and $\phi$ are identified from the initial data $u_0$ and $v_0$. The solution is therefore interpreted as two waves with constant velocity $c$ moving in opposite directions along the $x$-axis. In the particular case that $v_0(x) = 0$, the solution is given by $u(x,t) = (u_0(x+ct) + u_0(x-ct))/2$. It follows that the solution may not always be represented in a separated form with respect to both space and time depending on the choice of $u_0$.

We nevertheless provide the analytical solution in the case of a simple problem, that is, taking $f(x,t) = 0$, $v_0(x) = 0$, $g(t) = 0$, and $c$ constant. Moreover, the initial condition on the displacement is chosen as $u_0(x) = Fx/(EA)$, which corresponds to the equilibrium state of the bar when subjected to a force $F$ at $x = \ell$. The displacement $u$ satisfies the following system of equations:

$$\begin{aligned} \ddot{u} - c^2 u'' &= 0, && \forall (x,t) \in \Omega \times \mathcal{I}, \\ u(x,0) &= u_0(x), && \forall x \in \Omega, \\ \dot{u}(x,0) &= 0, && \forall x \in \Omega, \\ u(0,t) &= 0, && \forall t \in \mathcal{I}, \\ EAu'(\ell,t) &= 0, && \forall t \in \mathcal{I}. \end{aligned} \tag{2.6}$$

Using the method of separation of variables, we search for solutions in the separated form:

$$u(x,t) = \chi(x)\psi(t).$$

Substituting the above expression for $u$ in (2.6) yields:

$$\frac{\chi''(x)}{\chi(x)} = \frac{1}{c^2}\frac{\ddot{\psi}(t)}{\psi(t)}, \qquad \forall (x,t) \in \Omega \times \mathcal{I}.$$

It follows that one has to find constants $\lambda \in \mathbb{R}$ such that the function $\chi(x)$ satisfies the eigenvalue problem:

$$\begin{aligned} \chi''(x) + \lambda\chi(x) &= 0, && \forall x \in \Omega, \\ \chi(0) &= 0, \\ \chi'(\ell) &= 0, \end{aligned} \tag{2.7}$$

and such that the function $\psi(t)$ satisfies the ordinary differential equation:

$$\ddot{\psi}(t) + c^2\lambda\psi(t) = 0, \qquad \forall t \in \mathcal{I}. \tag{2.8}$$

The solutions to the eigenvalue problem (2.7) consist of the eigenvalues $\lambda_k$ and associated eigenfunctions $\chi_k$:

$$\lambda_k = \left[\frac{(2k-1)\pi}{2\ell}\right]^2, \qquad \chi_k(x) = \sin\left(\sqrt{\lambda_k}x\right), \qquad \forall k = 1, 2, \dots$$

while the solution to (2.8) for each eigenvalue $\lambda_k$ is given as:

$$\psi_k(t) = \alpha_k \cos\left(c\sqrt{\lambda_k}t\right) + \beta_k \sin\left(c\sqrt{\lambda_k}t\right).$$

The general solution to the problem thus reads:

$$u(x,t) = \sum_{k=1}^{+\infty} \chi_k(x)\psi_k(t) = \sum_{k=1}^{+\infty} \sin\frac{(2k-1)\pi x}{2\ell}\left[\alpha_k \cos\frac{(2k-1)\pi ct}{2\ell} + \beta_k \sin\frac{(2k-1)\pi ct}{2\ell}\right].$$

Using the initial condition on the velocity, i.e. $\dot{u}(x,0) = v_0(x) = 0$, implies that $\beta_k = 0$ for all $k = 1, 2, \dots$ Moreover, the coefficients $\alpha_k$ correspond to the coefficients of the sine Fourier series associated with the initial displacement $u_0(x) = Fx/(EA)$. It follows that the displacement field reads:

$$u(x,t) = \frac{8F\ell}{\pi^2 EA}\sum_{k=1}^{+\infty}\frac{(-1)^{k+1}}{(2k-1)^2}\sin\frac{(2k-1)\pi x}{2\ell}\cos\frac{(2k-1)\pi ct}{2\ell}, \qquad \forall (x,t) \in \bar{\Omega}\times\bar{\mathcal{I}}. \tag{2.9}$$

We observe in this case that the solution can be represented in a separated form and that the coefficients of the series decrease with a quadratic rate. We shall use this analytical solution to assess the accuracy of our calculations in some of the numerical examples.

## 2.2 Weak formulations of the problem

The construction of weak formulations of the problem is not unique. We present below two formulations based on the Lagrangian and the Hamiltonian approaches. We first recall the Hamiltonian's principle that will be used in the derivation.

### 2.2.1 Hamilton's Weak Principle

Let $q$ denote the generalized coordinates of the system, corresponding here to the displacement field $u$, and let $t_A$ and $t_B$ be two specified times. We note that the principle was originally stated assuming that the initial and final states of the system under study were known, $q(x, t_A) = q_A(x)$ and $q(x, t_B) = q_B(x)$. Given a Lagrangian functional $\mathcal{L}$ of the system, the action functional of $q$, denoted by $\mathcal{S}[q]$, is defined as [35, 36]:

$$\mathcal{S}[q] = \int_{t_A}^{t_B} \mathcal{L}(q(t), \dot{q}(t), t)\, dt. \tag{2.10}$$

The Hamilton's Weak Principle states that the evolution of $q$ followed by the physical system between the states $q_A$ and $q_B$ is a stationary point of the action functional:

$$\mathcal{S}'[q](q^*) = 0, \qquad \forall q^* \in V_0, \tag{2.11}$$

where $V_0$ is the space of perturbations $q^*$ that vanish at $t_A$ and $t_B$. The precise definition of $V$ depends on the choice of the Lagrangian. Here, $\mathcal{S}'[q](q^*)$ is the Gâteaux derivative of $\mathcal{S}[q]$ defined at $q$ with respect to the perturbation $q^*$, i.e.

$$\mathcal{S}'[q](q^*) = \lim_{\theta \to 0} \frac{1}{\theta}\Big(\mathcal{S}[q + \theta q^*] - \mathcal{S}[q]\Big).$$

In the particular case where the states $q$ at times $t_A$ and $t_B$ are unknown, the principle of least action can be recast as [84]:

$$\mathcal{S}'[q](q^*) = \left[\frac{\partial \mathcal{L}}{\partial \dot{q}} q^*\right]_{t_A}^{t_B}, \qquad \forall q^* \in V. \tag{2.12}$$

We note here that the perturbations $q^*$ in $V$ do not necessarily vanish at $t_A$ or $t_B$. Later in the manuscript, all our test cases consider the initial displacement $q_A$ to be known while $q_B$ remains unknown.

### 2.2.2 The Lagrangian formalism

The evolution of the generalized coordinates function $q$, describing the displacement as a function of $x$ and $t$, defines a so-called trajectory of a system in the configuration space. The trajectory of a physical system is thus entirely determined by the knowledge of $q$. The objective of Joseph-Louis Lagrange in his seminal treatise *Méchanique Analitique* first published in 1788 [77] was to lay down, once and for all, the foundations of analytical mechanics. In fact, he introduced as early as 1756 the action functional $\mathcal{S}$ as defined in (2.10).

### Continuous formulation

For our problem (2.1)-(2.5), the Lagrangian functional $\mathcal{L}$ reads:

$$\mathcal{L}(q,\dot{q},t) = \underbrace{\frac{1}{2}\int_\Omega \rho A \dot{q}^2\,dx}_{\text{Kinetic Energy}} - \underbrace{\frac{1}{2}\int_\Omega EA\left(\frac{\partial q}{\partial x}\right)^2 dx}_{\text{Potential Energy}} + \underbrace{\int_\Omega fq\;dx + g(t)q(\ell,t)}_{\text{External Energy}}, \tag{2.13}$$

where the field $q$ satisfies the initial conditions (2.2)-(2.3). In other words, the space of trial fields $q$ is:

$$\begin{aligned}\mathcal{U}_L = \{&q \in L^2(\mathcal{I}, H^1(\Omega)) \cap H^1(\mathcal{I}, L^2(\Omega));\\ &\qquad q(0,t) = 0,\ \forall t \in \mathcal{I};\ q(x,0) = u_0(x),\ \dot{q}(x,0) = v_0(x),\ \forall x \in \Omega\}.\end{aligned} \tag{2.14}$$

Requiring that the trajectory $q$ be a stationary point of the action functional $\mathcal{S}$ leads to the so-called Euler-Lagrange equations. Using the Lagrangian (2.13), the Gâteaux derivative of $\mathcal{S}$ is given by:

$$\begin{aligned}\mathcal{S}'[q](q^*) &= \lim_{\theta\to 0}\frac{1}{\theta}\left[\int_\mathcal{I}\mathcal{L}(q+\theta q^*, \dot{q}+\theta\dot{q}^*, t)\,dt - \int_\mathcal{I}\mathcal{L}(q,\dot{q},t)\,dt\right]\\ &= \lim_{\theta\to 0}\frac{1}{\theta}\int_\mathcal{I}\Big[\mathcal{L}(q+\theta q^*, \dot{q}+\theta\dot{q}^*, t) - \mathcal{L}(q,\dot{q},t)\Big]dt\\ &= \int_\mathcal{I}\int_\Omega \rho A\dot{q}^*\dot{q} - EA\frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x} + q^* f\,dxdt + \int_\mathcal{I} q^*(\ell,t)g(t)\,dt,\end{aligned}$$

where the space of perturbations $q^*$ is given by:

$$\begin{aligned}\mathcal{V}_L = \{&q^* \in L^2(\mathcal{I}, H^1(\Omega)) \cap H^1(\mathcal{I}, L^2(\Omega));\\ &\qquad q^*(0,t) = 0,\ \forall t \in \mathcal{I};\ q^*(x,0) = 0,\ \dot{q}^*(x,0) = 0,\ \forall x \in \Omega\}.\end{aligned} \tag{2.15}$$

Then, using (2.12), we obtain the equation:

$$\int_\mathcal{I}\int_\Omega \rho A\dot{q}^*\dot{q} - EA\frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x} + q^* f\,dxdt + \int_\mathcal{I} q^*(\ell,t)g(t)\;dt = \left[\int_\Omega \rho A q^*\dot{q}\,dx\right]_0^T, \quad \forall q^* \in \mathcal{V}_L.$$

It follows that a weak formulation of the problem reads:

Find $q \in \mathcal{U}_L$ such that

$$\begin{aligned}\int_{\mathcal{I}}\int_{\Omega} \rho A \dot{q}^* \dot{q} - EA \frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x}\, dxdt - \int_{\Omega} \rho A q^*(x,T)\dot{q}(x,T)\, dx \\ = -\int_{\mathcal{I}}\int_{\Omega} q^* f\, dxdt - \int_{\mathcal{I}} q^*(\ell,t) g(t)\, dt, \quad \forall q^* \in \mathcal{V}_L.\end{aligned} \tag{2.16}$$

We note that the above formulation is equivalent to the strong form (2.1)-(2.5) of the problem for sufficiently smooth data. Indeed, integration by parts with respect to the time variable yields:

$$\int_{\mathcal{I}}\int_{\Omega} \rho A q^* \ddot{q} + EA \frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x}\, dxdt = \int_{\mathcal{I}}\int_{\Omega} q^* f\, dxdt + \int_{\mathcal{I}} q^*(\ell,t) g(t)\, dt, \quad \forall q^* \in \mathcal{V}_L. \tag{2.17}$$

Moreover, following an integration by parts with respect to the space variable, one obtains:

$$\int_{\mathcal{I}}\int_{\Omega} q^* \left(\rho A \ddot{q} - EA \frac{\partial^2 q}{\partial x^2} - f\right) dxdt + \int_{\mathcal{I}} q^*(\ell,t) \left(EA \frac{\partial q}{\partial x}(\ell,t) - g(t)\right) dt = 0, \quad \forall q^* \in \mathcal{V}_L,$$

which allows us to recover the strong form of the wave equation (2.1) and the Neumann boundary condition (2.5).

### Discrete formulation

The objective here is to define the discrete problem using a Finite Element method in space and a finite difference approach in time. In order to do so, we consider the semi-weak formulation instead of the weak formulation (2.16):

Find $q(\cdot,t) \in V$, $\forall t \in \mathcal{I}$, such that

$$\begin{aligned}\int_{\Omega} \rho(x) A q^*(x) \ddot{q}(x,t) + EA \frac{\partial q^*}{\partial x}(x) \frac{\partial q}{\partial x}(x,t)\, dx \\ = \int_{\Omega} q^*(x) f(x,t)\, dx + q^*(\ell) g(t), \quad \forall q^* \in V, \quad \forall t \in \mathcal{I}\end{aligned}$$

and that satisfies the initial conditions

$$\begin{aligned} q(x,0) &= u_0(x), \quad \forall x \in \Omega, \\ \dot{q}(x,0) &= v_0(x), \quad \forall x \in \Omega, \end{aligned}$$

where $V$ is the vector space of functions defined on $\Omega$ as:

$$V = \left\{ v \in H^1(\Omega) : \; v(0) = 0 \right\}.$$

We partition the domain into $N_e$ elements, denoted by $K_e$, such that $\overline{\Omega} = \cup_{e=1}^{N_e} K_e$ and $\mathrm{Int}(K_i) \cap \mathrm{Int}(K_j) = \varnothing$, $\forall i, j = 1, \dots, N_e$, $i \neq j$. We then associate with the mesh a finite element space $V^h \subset V$, $\dim V^h = n$, based on continuous piecewise polynomial functions defined on $\Omega$:

$$V^h = \{v_h \in V : \; v_h|_{K_e} \in \mathbb{P}_k(K_e), \; e = 1, \dots, N_e\},$$

where $\mathbb{P}_k(K_e)$ denotes the space of polynomial functions of degree $k$ on $K_e$. Let $\phi_i$, with $i = 1, \dots, n$, denote the basis functions of $V^h$, i.e. $V^h = \mathrm{span}\{\phi_i\}$. We then search for finite element solutions in the form:

$$q_h(x,t) = \sum_{j=1}^{n} q_j(t)\phi_j(x)$$

where the degrees of freedom $q_j$ depend on time. The Finite Element problem using the Galerkin method thus reads:

Find $q_h(\cdot, t) \in V^h$, $\forall t \in \mathcal{I}$, such that

$$\begin{aligned} \int_\Omega \rho(x) A \phi_i(x) \ddot{q}_h(x,t) + EA \frac{\partial \phi_i}{\partial x}(x) \frac{\partial q_h}{\partial x}(x,t)\, dx & \\ = \int_\Omega \phi_i(x) f(x,t)\, dx + \phi_i(\ell) g(t), &\quad \forall i = 1, \dots, n, \quad \forall t \in \mathcal{I} \end{aligned}$$

and that satisfies the initial conditions

$$\begin{aligned} q_h(x,0) &= u_{0,h}(x), \quad \forall x \in \Omega, \\ \dot{q}_h(x,0) &= v_{0,h}(x), \quad \forall x \in \Omega, \end{aligned}$$

where $u_{0,h}$ and $v_{0,h}$ are interpolants or projections of $u_0$ and $v_0$ in the space $V^h$. The above problem can be recast in compact form as:

$$\begin{aligned} M\ddot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) &= \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I} \\ \boldsymbol{q}(0) &= \boldsymbol{u}_0, \\ \dot{\boldsymbol{q}}(0) &= \boldsymbol{v}_0, \end{aligned} \tag{2.18}$$

where $M$ and $K$ are the global mass and stiffness matrices, respectively, both being symmetric and positive definite:

$$M_{ij} = \int_\Omega \rho A \phi_i \phi_j \, dx, \qquad K_{ij} = \int_\Omega EA \phi_i' \phi_j' \, dx, \qquad \forall i, j = 1, \dots, n,$$

$\boldsymbol{f}(t)$ is the loading vector at time $t$:

$$f_i(t) = \int_\Omega \phi_i(x) f(x,t)\,dx + \phi_i(\ell) g(t), \qquad \forall i = 1,\dots,n,$$

$\boldsymbol{q}(t)$ is the vector of degrees of freedom:

$$\boldsymbol{q}(t) = \begin{bmatrix} q_1(t) & \dots & q_n(t) \end{bmatrix}^T$$

and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial vectors:

$$\begin{aligned} \boldsymbol{u}_0 &= \begin{bmatrix} u_{0,1} & \dots & u_{0,n} \end{bmatrix}^T, \\ \boldsymbol{v}_0 &= \begin{bmatrix} v_{0,1} & \dots & v_{0,n} \end{bmatrix}^T. \end{aligned}$$

A classical approach [80,81] to discretize in time the system of second-order differential equations (2.18) consists first in rewriting the system as a system of first-order differential equations by introducing the vector of velocities $\boldsymbol{w} = \dot{\boldsymbol{q}}$, i.e.

$$\dot{\boldsymbol{q}}(t) - \boldsymbol{w}(t) = 0, \qquad \forall t \in \mathcal{I}, \tag{2.19}$$

$$M\dot{\boldsymbol{w}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{2.20}$$

and then in applying the Crank-Nicolson scheme (also referred to as the implicit trapezoidal rule) to both (2.19) and (2.20). Dividing the time domain $\mathcal{I}$ into $n_t$ subintervals $\mathcal{I}^n = [t^n, t^{n+1}]$, $n = 1,\dots,n_t$ of size $h_t = t^{n+1} - t^n$, we evaluate $\boldsymbol{q}^n$ and $\boldsymbol{w}^n$, $n = 0,\dots,n_t$, such that $\boldsymbol{q}^0 = \boldsymbol{u}_0$ and $\boldsymbol{w}^0 = \boldsymbol{v}_0$, and:

$$\begin{aligned} 2(\boldsymbol{q}^{n+1} - \boldsymbol{q}^n) - h_t(\boldsymbol{w}^n + \boldsymbol{w}^{n+1}) &= 0, && \forall n = 0,\dots,n_t - 1, \\ 2M(\boldsymbol{w}^{n+1} - \boldsymbol{w}^n) + h_t K(\boldsymbol{q}^n + \boldsymbol{q}^{n+1}) &= h_t(\boldsymbol{f}^n + \boldsymbol{f}^{n+1}), && \forall n = 0,\dots,n_t - 1. \end{aligned}$$

The above system of equations can be conveniently recast in matrix form as:

$$\begin{bmatrix} h_t K & 2M \\ 2M & -h_t M \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{n+1} \\ \boldsymbol{w}^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t K & 2M \\ 2M & h_t M \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^n \\ \boldsymbol{w}^n \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{f}^n + \boldsymbol{f}^{n+1} \\ 0 \end{bmatrix}, \quad \forall n = 0,\dots,n_t - 1, \tag{2.21}$$

where we have multiplied the first row by matrix $M$. It is worth noting that the scheme is not only implicit and second-order, but preserves the energy of the system over time [80–82]. We shall compare the scheme to that obtained using the Hamiltonian framework presented below.

### 2.2.3 The Hamiltonian formalism

We recall that Lagrange described the evolution of a system in terms of the generalized coordinates $q$, and implicitly of its first derivative $\dot{q}$, in the configuration space. Hamilton extended the work of Lagrange in 1834 [35] by describing the evolution of the system in the phase space, introducing the generalized coordinates $q$ and their generalized (or conjugate) momenta $p = \rho A \dot{q}$ as independent quantities. In order to do so, he applied a Legendre transform to the Lagrangian with respect to $\dot{q}$ (with $q$ fixed) and thus defined the Hamiltonian functional $\mathcal{H}$ as [76]:

$$\mathcal{H}(q,p,t) = \int_{\mathcal{I}} p\dot{q}\, dt - \mathcal{L}(q,\dot{q},t). \tag{2.22}$$

While the Lagrangian is written in terms of a difference between the kinetic energy and the potential energy, the Hamiltonian corresponds to the sum of these two energies. In fact, it actually represents the total energy of the system under study in the case of conservative systems, see below.

#### Continuous formulation

The action $\mathcal{S}$ (2.10) is defined in terms of the Hamiltonian functional (2.22) as:

$$\mathcal{S}[q,p] = \int_{\mathcal{I}} \dot{q}p - \mathcal{H}(q,p,t)\, dt$$

where the Hamiltonian functional for our problem reads:

$$\mathcal{H}(q,p,t) = \underbrace{\frac{1}{2}\int_{\Omega} \frac{1}{\rho A} p^2\, dx}_{\text{Kinetic Energy}} + \underbrace{\frac{1}{2}\int_{\Omega} EA \left(\frac{\partial q}{\partial x}\right)^2 dx}_{\text{Potential Energy}} - \underbrace{\left(\int_{\Omega} fq\, dx + g(t)q(\ell,t)\right)}_{\text{External Energy}}$$

The generalized coordinate and momenta are searched in the spaces:

$$\begin{aligned}
\mathcal{U}_H &= \{q \in L^2(\mathcal{I}; H^1(\Omega));\ q(0,t) = 0,\ \forall t \in \mathcal{I};\ q(x,0) = u_0(x),\ \forall x \in \Omega\},\\
\mathcal{W}_H &= \{p \in L^2(\mathcal{I}; L^2(\Omega));\ p(x,0) = \rho A v_0(x),\ \forall x \in \Omega\}.
\end{aligned}$$

The Hamilton's Weak Principle then states that the trajectory $(q,p)$ of the system in the phase space should satisfy:

$$\mathcal{S}'[q,p](q^*,p^*) = \left[\int_{\Omega} q^* p\, dx\right]_0^T,$$

where $\mathcal{S}'[q,p](q^*,p^*)$ denotes here the Gâteaux derivative of $\mathcal{S}[q,p]$ with respect to a perturbation $(q^*,p^*)$ belonging to the spaces:

$$\begin{aligned}
\mathcal{V}_H &= \{q^* \in L^2(\mathcal{I};H^1(\Omega));\ q^*(0,t)=0,\ \forall t\in\mathcal{I};\ q^*(x,0)=0,\ \forall x\in\Omega\},\\
\mathcal{Z}_H &= \{p^* \in L^2(\mathcal{I};L^2(\Omega));\ p^*(x,0)=0,\ \forall x\in\Omega\}.
\end{aligned}$$

We easily compute the Gâteaux derivative of $\mathcal{S}[q,p]$ as:

$$\mathcal{S}'[q,p](q^*,p^*) = \int_{\mathcal{I}}\int_{\Omega} \dot{q}^*p + p^*\dot{q} - \frac{1}{\rho A}p^*p - EA\frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x} + q^*f\,dxdt\ + \int_{\mathcal{I}} q^*(\ell,t)g(t)\,dt,$$

so that a weak formulation of the problem reads:

Find $(q,p)\in\mathcal{U}_H\times\mathcal{W}_H$ such that

$$\begin{aligned}
\int_{\mathcal{I}}\int_{\Omega} \dot{q}^*p + p^*\dot{q} - \frac{1}{\rho A}p^*p - EA\frac{\partial q^*}{\partial x}\frac{\partial q}{\partial x}dxdt\ - \int_{\Omega} q^*(T)p(T)\,dx& \\
= -\int_{\mathcal{I}}\int_{\Omega} q^*f\,dxdt\ - \int_{\mathcal{I}} q^*(\ell,t)g(t)\,dt, \qquad \forall(q^*,p^*)\in\mathcal{V}_H\times\mathcal{Z}_H.&
\end{aligned} \tag{2.23}$$

Integrating by parts with respect to time and space for sufficiently smooth data, we can rewrite (2.23) as:

$$\begin{aligned}
\int_{\mathcal{I}}\int_{\Omega} p^*\left(\frac{1}{\rho A}p - \dot{q}\right) + q^*\left(\dot{p} - EA\frac{\partial^2 q}{\partial x^2} - f\right)dxdt& \\
+\int_{\mathcal{I}} q^*(\ell,t)\left(EA\frac{\partial q}{\partial x}(\ell,t) - g(t)\right)dt = 0, \qquad \forall(q^*,p^*)\in\mathcal{V}_H\times\mathcal{Z}_H,&
\end{aligned}$$

or, in a decoupled fashion with respect to the test functions $q^*$ and $p^*$, as:

$$\begin{aligned}
&\int_{\mathcal{I}}\int_{\Omega} p^*\left(\frac{1}{\rho A}p - \dot{q}\right)dxdt = 0, && \forall p^*\in\mathcal{Z}_H,\\
&\int_{\mathcal{I}}\int_{\Omega} q^*\left(\dot{p} - EA\frac{\partial^2 q}{\partial x^2} - f\right)dxdt\ + \int_{\mathcal{I}} q^*(\ell,t)\left(EA\frac{\partial q}{\partial x}(\ell,t) - g(t)\right)dt = 0, && \forall q^*\in\mathcal{V}_H.
\end{aligned}$$

The Hamiltonian formulation (2.23) can be viewed as a mixed problem for which the coupled differential equations in strong form are given by:

$$\begin{aligned}
\dot{q} &= \frac{1}{\rho A}p, && \forall x\in\Omega, \forall t\in\mathcal{I},\\
\dot{p} &= EA\frac{\partial^2 q}{\partial x^2} + f, && \forall x\in\Omega, \forall t\in\mathcal{I}.
\end{aligned} \tag{2.24}$$

We observe that the system of equations is equivalent to (2.1) by introducing the auxiliary variable $p = \rho A\dot{q}$. The system of equations (2.24) is usually referred to as the canonical Hamilton equations. One advantage of the Hamiltonian formalism is that it explicitly informs one on how to define this auxiliary variable.

### Discrete formulation

For convenience, we first recast the weak formulation (2.23) as the system of equations:

$$\int_{\mathcal{I}} \int_{\Omega} p^* \left( \frac{1}{\rho A} p - \dot{q} \right) dxdt = 0, \qquad \forall p^* \in \mathcal{Z}_H, \tag{2.25}$$

$$\int_{\mathcal{I}} \int_{\Omega} q^* \dot{p} + EA \frac{\partial q^*}{\partial x} \frac{\partial q}{\partial x} \, dxdt = \int_{\mathcal{I}} \int_{\Omega} q^* f \, dxdt \; + \int_{\mathcal{I}} q^*(\ell, t) g(t) \, dt, \qquad \forall q^* \in \mathcal{V}_H. \tag{2.26}$$

The objective here is to discretize the set of equations using a Finite Element method in both space and time. For the spatial discretization, we employ the same mesh and FE space $V^h$ for $q$ and $p$ as the ones used in the Lagrangian formulation. In other words, we search for Finite Element solutions in the form:

$$\begin{aligned} q_h(x,t) &= \sum_{j=1}^{n} q_j(t) \phi_j(x), \\ p_h(x,t) &= \sum_{j=1}^{n} p_j(t) \phi_j(x), \end{aligned}$$

and denote by $\boldsymbol{q}$ and $\boldsymbol{p}$ the vectors of time-dependent degrees of freedom $q_j$ and $p_j$, $j = 1, \ldots, n$, respectively. In the same manner, we consider test functions in the form:

$$\begin{aligned} q^*_h(x,t) &= \sum_{i=1}^{n} q^*_i(t) \phi_i(x), \\ p^*_h(x,t) &= \sum_{i=1}^{n} p^*_i(t) \phi_i(x). \end{aligned}$$

Inserting the trial and test functions in (2.25) and (2.26), one obtains the semi-discrete set of equations:

$$\int_{\mathcal{I}} \boldsymbol{p}^{*T}(t) \left( \bar{\bar{M}} \boldsymbol{p}(t) - \bar{M} \dot{\boldsymbol{q}}(t) \right) dt = 0, \tag{2.27}$$

$$\int_{\mathcal{I}} \boldsymbol{q}^{*T}(t) \left( \bar{M} \dot{\boldsymbol{p}}(t) + K \boldsymbol{q}(t) \right) dt = \int_{\mathcal{I}} \boldsymbol{q}^{*T}(t) \boldsymbol{f}(t) \, dt, \tag{2.28}$$

where the matrices $\bar{M}$ and $\bar{\bar{M}}$ are the symmetric positive-definite matrices:

$$\bar{M}_{ij} = \int_\Omega \phi_i \phi_j \, dx, \qquad \bar{\bar{M}}_{ij} = \int_\Omega \frac{1}{\rho A} \phi_i \phi_j \, dx, \qquad \forall i, j = 1, \dots, n.$$

In order to approximate the functions $\boldsymbol{q}$ and $\boldsymbol{p}$ with respect to time, we follow an approach similar to the one proposed in [76]. We thus consider continuous piecewise linear trial functions on each subinterval $\mathcal{I}^n = [t^n, t^{n+1}]$, $n = 0, \dots, n_t - 1$:

$$\begin{aligned} \boldsymbol{q}(t) &\approx \left[\frac{t^{n+1} - t}{h_t}\right] \boldsymbol{q}^n + \left[\frac{t - t^n}{h_t}\right] \boldsymbol{q}^{n+1} \\ \boldsymbol{p}(t) &\approx \left[\frac{t^{n+1} - t}{h_t}\right] \boldsymbol{p}^n + \left[\frac{t - t^n}{h_t}\right] \boldsymbol{p}^{n+1} \end{aligned}$$

and piecewise constant test functions on each subinterval $\mathcal{I}^n$, i.e.

$$\boldsymbol{q}^*(t) = \boldsymbol{q}^{*n}, \qquad \boldsymbol{p}^*(t) = \boldsymbol{p}^{*n}.$$

Using the above expressions in (2.27) and (2.28) gives:

$$\begin{aligned} &\sum_{n=0}^{n_t-1} [\boldsymbol{p}^{*n}]^T \int_{\mathcal{I}^n} \bar{\bar{M}} \left( \left[\frac{t^{n+1} - t}{h_t}\right] \boldsymbol{p}^n + \left[\frac{t - t^n}{h_t}\right] \boldsymbol{p}^{n+1} \right) - \bar{M} \left( \frac{\boldsymbol{q}^{n+1} - \boldsymbol{q}^n}{h_t} \right) dt = 0, \\ &\sum_{n=0}^{n_t-1} [\boldsymbol{q}^{*n}]^T \int_{\mathcal{I}^n} \bar{M} \left( \frac{\boldsymbol{p}^{n+1} - \boldsymbol{p}^n}{h_t} \right) + K \left( \left[\frac{t^{n+1} - t}{h_t}\right] \boldsymbol{q}^n + \left[\frac{t - t^n}{h_t}\right] \boldsymbol{q}^{n+1} \right) dt \\ &\qquad\qquad = \sum_{n=0}^{n_t-1} [\boldsymbol{q}^{*n}]^T \int_{\mathcal{I}^n} \left[\frac{t^{n+1} - t}{h_t}\right] \boldsymbol{f}^n + \left[\frac{t - t^n}{h_t}\right] \boldsymbol{f}^{n+1} dt, \end{aligned}$$

which, after integration, yields:

$$\begin{aligned} &\sum_{n=0}^{n_t-1} [\boldsymbol{p}^{*n}]^T \left[ \frac{h_t}{2} \bar{\bar{M}} \Big( \boldsymbol{p}^n + \boldsymbol{p}^{n+1} \Big) - \bar{M} \Big( \boldsymbol{q}^{n+1} - \boldsymbol{q}^n \Big) \right] = 0, \\ &\sum_{n=0}^{n_t-1} [\boldsymbol{q}^{*n}]^T \left[ \bar{M} \Big( \boldsymbol{p}^{n+1} - \boldsymbol{p}^n \Big) + \frac{h_t}{2} K \Big( \boldsymbol{q}^n + \boldsymbol{q}^{n+1} \Big) - \frac{h_t}{2} \Big( \boldsymbol{f}^n + \boldsymbol{f}^{n+1} \Big) \right] = 0. \end{aligned}$$

Since the above equations hold for any arbitrary test fields, the problem consists in solving for $\boldsymbol{q}^n$ and $\boldsymbol{p}^n$, $n = 0, \dots, n_t$, such that $\boldsymbol{q}^0 = \boldsymbol{u}_0$ and $\boldsymbol{p}^0 = \rho A \boldsymbol{v}_0$, and:

$$\begin{aligned} &2\bar{M}\boldsymbol{q}^{n+1} - h_t \bar{\bar{M}} \boldsymbol{p}^{n+1} = 2\bar{M}\boldsymbol{q}^n + h_t \bar{\bar{M}} \boldsymbol{p}^n, && \forall n = 0, \dots, n_t - 1, \\ &h_t K \boldsymbol{q}^{n+1} + 2\bar{M}\boldsymbol{p}^{n+1} = -h_t K \boldsymbol{q}^n + 2\bar{M}\boldsymbol{p}^n + h_t \Big( \boldsymbol{f}^n + \boldsymbol{f}^{n+1} \Big), && \forall n = 0, \dots, n_t - 1, \end{aligned}$$

which can be conveniently recast in matrix form as:

$$\begin{bmatrix} h_t K & 2\bar{M} \\ 2\bar{M} & -h_t\bar{\bar{M}} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{n+1} \\ \boldsymbol{p}^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t K & 2\bar{M} \\ 2\bar{M} & h_t\bar{\bar{M}} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{n} \\ \boldsymbol{p}^{n} \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{f}^{n} + \boldsymbol{f}^{n+1} \\ 0 \end{bmatrix}, \quad \forall n = 0, \dots, n_t - 1. \tag{2.29}$$

We note that we would have arrived exactly at the same set of equations if we had chosen to discretize the problem in time by the finite differences Crank-Nicolson scheme. More interestingly, we observe that (2.29) has exactly the same structure as (2.21) except for the fact that matrix $M$ in (2.21) has been replaced by either $\bar{M}$ or $\bar{\bar{M}}$. We shall see in the following how this slight difference will affect the results within the PGD framework.

Finally, we can conveniently collect the degrees of freedom of the FE solution into the matrix $U$ of size $n \times n_t$:

$$U = \begin{bmatrix} \boldsymbol{q}^1 & \boldsymbol{q}^2 & \dots & \boldsymbol{q}^{n_t} \end{bmatrix} = \begin{bmatrix} q_1^1 & q_1^2 & \dots & q_1^{n_t} \\ q_2^1 & q_2^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ q_n^1 & \dots & \dots & q_n^{n_t} \end{bmatrix}.$$

We will use the solutions given by (2.21) and (2.29) as reference solutions when assessing the results of the PGD. In particular, we will perform some Singular Value Decomposition of $U$ to identify the principal components or modes of the FE solution.

### 2.3 PGD reduced-order modeling

The PGD framework aims at searching for an approximation of the given field of interest, e.g. the generalized coordinate $q$, in the separated form:

$$q(x,t) \approx q_m(x,t) = q_0(x,t) + \sum_{i=1}^{m} \lambda_i(t)\mu_i(x),$$

where the truncation parameter $m$ denotes the number of modes in the representation, the $\lambda_i$'s and the $\mu_i$'s stand for the temporal and spatial modes, respectively. $q_0$ is a lift function that satisfies the non-homogeneous Dirichlet boundary conditions and initial conditions so that each mode in the separated representation satisfies the corresponding homogeneous boundary and initial conditions [85].

The PGD solution is often computed based on a greedy algorithm [49, 85]. Assuming that the $q_{m-1}$ mode has been computed, the approach consists then in finding the $m^{\text{th}}$ enrichment mode as follows:

$$q_m(x,t) = q_{m-1}(x,t) + \lambda(t)\mu(x),$$

where the subscript $m$ has been dropped from $\lambda_m$ and $\mu_m$ for the sake of clarity in the notation. Inserting the trial solution $q_m$ in the governing differential equations using the Galerkin method or a residual minimization approach [49, 85, 86] leads to the solution of a non-linear system for the unknown functions $\lambda$ and $\mu$. The problem is usually solved by means of an appropriate iterative scheme, such as a fixed-point algorithm that will be considered here, in which one determines in an alternating fashion at each iteration of the algorithm the solution $\mu$ with $\lambda$ known and then $\lambda$ with $\mu$ known [85, 87].

We describe below the derivation of the PGD formulation using the Lagrangian and Hamiltonian formalism. We will construct first the formulations at the continuous level and then propose some numerical schemes to discretize the problems.

### 2.3.1 Lagrangian-based PGD

We consider first the Lagrangian framework. We start from the formulation (2.17) as we will use a Finite Element approach in space and a Finite Differences approach in time.

#### Continuous formulation

Substituting $q_{m-1}(x,t) + \lambda(t)\mu(x)$ for $q_m(x,t)$ in (2.17), one gets:

$$\begin{aligned}&\int_{\mathcal{I}}\int_{\Omega} \rho A q^* \ddot{\lambda}\mu + EA\frac{\partial q^*}{\partial x}\lambda\mu' \, dxdt \\ &\qquad = \int_{\mathcal{I}}\int_{\Omega} q^*(f - \rho A\ddot{q}_{m-1}) - EA\frac{\partial q^*}{\partial x}\frac{\partial q_{m-1}}{\partial x}\, dxdt + \int_{\mathcal{I}} q^*(\ell,t)g(t)\, dt, \qquad \forall q^* \in \mathcal{V}_L,\end{aligned} \tag{2.30}$$

where the vector space $\mathcal{V}_L$ is defined as in (2.15). In view of using a fixed-point approach, we now derive the problems for $\mu$ and for $\lambda$.

We assume first that $\lambda$ is known and search for $\mu \in V$. We thus choose test functions in the form $q^*(x,t) = \lambda(t)\mu^*(x)$ with $\mu^* \in V$. Equation (2.30) thus reduces to:

$$\int_{\Omega} m_{\ell t}\mu\mu^* + k_{\ell t}\mu'\mu^{*\prime} dx = \int_{\Omega} r_{\ell\mu}(\mu^*)\, dx + \left(\int_{\mathcal{I}} \lambda(t)g(t)\, dt\right)\mu^*(\ell), \qquad \forall \mu^* \in V, \tag{2.31}$$

with:

$$
\begin{aligned}
m_{\ell t} &= \rho A \int_{\mathcal{I}} \ddot{\lambda}\lambda \, dt = \rho A \left( \dot{\lambda}(T)\lambda(T) - \int_{\mathcal{I}} \dot{\lambda}^2 \, dt \right), \\
k_{\ell t} &= EA \int_{\mathcal{I}} \lambda^2 \, dt, \\
r_{\ell \mu}(\mu^*) &= \left( \int_{\mathcal{I}} f\lambda \, dt \right) \mu^* - \rho A \left( \int_{\mathcal{I}} \ddot{q}_{m-1}\lambda \, dt \right) \mu^* - EA \left( \int_{\mathcal{I}} \frac{\partial q_{m-1}}{\partial x} \lambda \, dt \right) \mu^{*\prime},
\end{aligned}
$$

where $m_{\ell t}$, $k_{\ell t}$, and $r_{\ell \mu}(\mu^*)$ are possibly functions of the spatial variable only.

Similarly, we assume now that $\mu$ is known and search for $\lambda = \lambda(t)$. Choosing test functions in the form $q^*(x,t) = \mu(x)\lambda^*(t)$, Equation (2.30) then becomes:

$$
\int_{\mathcal{I}} \lambda^* \left( m_{\ell x}\ddot{\lambda} + k_{\ell x}\lambda \right) dt = \int_{\mathcal{I}} \lambda^* r_{\ell \lambda} \, dt, \qquad \forall \lambda^*, \tag{2.32}
$$

or simply:

$$
m_{\ell x}\ddot{\lambda}(t) + k_{\ell x}\lambda(t) = r_{\ell \lambda}(t), \qquad \forall t \in \mathcal{I}, \tag{2.33}
$$

with:

$$
\begin{aligned}
m_{\ell x} &= \int_{\Omega} \rho A \mu^2 \, dx, \\
k_{\ell x} &= \int_{\Omega} EA \left(\mu'\right)^2 dx, \\
r_{\ell \lambda}(t) &= \int_{\Omega} f\mu \, dx - \int_{\Omega} \rho A \ddot{q}_{m-1}\mu \, dx - \int_{\Omega} EA \frac{\partial q_{m-1}}{\partial x} \mu' \, dx + \mu(\ell) g(t),
\end{aligned}
$$

where $m_{\ell x}$ and $k_{\ell x}$ are constant.

### Discrete formulation

As before, we discretize (2.31) by the Finite Element method. In other words, we are looking for an approximate solution $\mu_h \in V^h$ of $\mu$:

$$
\mu(x) \approx \mu_h(x) = \sum_{j=1}^{n} \mu_j \phi_j(x),
$$

satisfying:

$$
\int_{\Omega} m_{\ell t}\mu_h\phi_i + k_{\ell t}\mu_h'\phi_i' dx = \int_{\Omega} r_{\ell \mu}(\phi_i) \, dx + \left( \int_{\mathcal{I}} \lambda g(t) \, dt \right) \phi_i(\ell), \qquad \forall i = 1, \ldots, n,
$$

which can be recast in matrix form as:

$$\Big(M_{\ell t} + K_{\ell t}\Big) U_{\ell\mu} = R_{\ell\mu}. \tag{2.34}$$

Here, the matrices $M_{\ell t}$ and $K_{\ell t}$ are the modified mass and stiffness matrices, respectively:

$$M_{\ell t} = M \left( \int_{\mathcal{I}} \ddot{\lambda}\lambda \; dt \right), \qquad K_{\ell t} = K \left( \int_{\mathcal{I}} \lambda^2 \; dt \right),$$

and the vector of degrees of freedom $U_{\ell\mu}$ and the loading vector $R_{\ell\mu}$ are given by:

$$U_{\ell\mu} = \begin{bmatrix} \vdots \\ \mu_i \\ \vdots \end{bmatrix}, \qquad R_{\ell\mu} = \begin{bmatrix} \vdots \\ \int_\Omega r_{\ell\mu}(\phi_i)\, dx + (\int_{\mathcal{I}} \lambda g(t)\, dt)\, \phi_i(\ell) \\ \vdots \end{bmatrix}.$$

For the time discretization, we again reduce the second-order equation (2.32) to a system of first-order equations by introducing the new variable $\omega = \dot{\lambda}$, i.e.:

$$\begin{aligned} \dot{\lambda}(t) - \omega(t) &= 0, & \forall t \in \mathcal{I}, \\ m_{\ell x}\dot{\omega}(t) + k_{\ell x}\lambda(t) &= r_{\ell\lambda}(t), & \forall t \in \mathcal{I}. \end{aligned}$$

and apply the Crank-Nicolson scheme to each equation. The scheme consists then in finding the pair $(\lambda_n, \omega_n) \in \mathbb{R}^2$, $n = 1, \ldots, n_t$ such that:

$$\begin{bmatrix} h_t k_{\ell x} & 2m_{\ell x} \\ 2m_{\ell x} & -h_t m_{\ell x} \end{bmatrix} \begin{bmatrix} \lambda^{n+1} \\ \omega^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t k_{\ell x} & 2m_{\ell x} \\ 2m_{\ell x} & h_t m_{\ell x} \end{bmatrix} \begin{bmatrix} \lambda^{n} \\ \omega^{n} \end{bmatrix} + h_t \begin{bmatrix} r_{\ell\lambda}^n + r_{\ell\lambda}^{n+1} \\ 0 \end{bmatrix}, \quad \forall n = 0, \ldots, n_t - 1, \tag{2.35}$$

where we have multiplied the first row by $m_{\ell x}$. We observe that the above system of equations has naturally the same structure as that in (2.21).

### 2.3.2 Hamiltonian-based PGD

The proper-generalized decomposition method applied within the Hamiltonian framework aims at approximating both the generalized coordinates $q$ and their generalized momenta $p$ in the separated form:

$$
\begin{aligned}
q(x,t) &\approx q_m(x,t) = q_{m-1}(x,t) + \lambda(t)\mu(x),\\
p(x,t) &\approx p_m(x,t) = p_{m-1}(x,t) + \omega(t)\nu(x).
\end{aligned}
$$

The goal in this section is to construct the problems that satisfy the enrichment modes $\lambda(t)\mu(x)$ and $\omega(t)\nu(x)$ assuming that both $q_{m-1}(x,t)$ and $p_{m-1}(x,t)$ have been calculated. We present first the continuous formulation of the problems.

### Continuous formulation

Replacing $q$ and $p$ in (2.23) by $q_m$ and $p_m$, respectively, one straightforwardly gets:

$$
\int_{\mathcal{I}}\int_{\Omega} p^*\left(\frac{1}{\rho A}\omega\nu - \dot{\lambda}\mu\right)dxdt = -\int_{\mathcal{I}}\int_{\Omega} p^*\left(\frac{1}{\rho A}p_{m-1} - \dot{q}_{m-1}\right)dxdt, \qquad \forall p^* \in \mathcal{Z}_H, \quad (2.36)
$$

$$
\begin{aligned}
\int_{\mathcal{I}}\int_{\Omega} q^*\dot{\omega}\nu + EA\frac{\partial q^*}{\partial x}\lambda\mu'\,dxdt = \int_{\mathcal{I}}\int_{\Omega} q^*(f - \dot{p}_{m-1}) - EA\frac{\partial q^*}{\partial x}\frac{\partial q_{m-1}}{\partial x}\,dxdt&\\
+\int_{\mathcal{I}} q^*(\ell,t)g(t)\,dt,& \qquad \forall q^* \in \mathcal{V}_H. \quad (2.37)
\end{aligned}
$$

As in the Lagrangian procedure, we first assume that $\lambda$ and $\omega$ are known and search for the solutions $\mu \in V$ and $\nu \in L^2(\Omega)$. We therefore choose test functions in the form $q^*(x,t) = \lambda(t)\mu^*(x)$ and $p^*(x,t) = \omega(t)\nu^*(x)$. Equations (2.36) and (2.37) thus become:

$$
\begin{aligned}
&\int_{\Omega} m_{ht}\nu\nu^* - c_{ht}\mu\nu^*\,dx = \int_{\Omega} r_{h\nu}(\nu^*)\,dx, && \forall \nu^* \in L^2(\Omega),\\
&\int_{\Omega} d_{ht}\nu\mu^* + k_{ht}\mu'\mu^{*\prime}dx = \int_{\Omega} r_{h\mu}(\mu^*)\,dx + \left(\int_{\mathcal{I}}\lambda g(t)\,dt\right)\mu^*(\ell), && \forall \mu^* \in V,
\end{aligned}
\qquad (2.38)
$$

with:

$$
\begin{aligned}
m_{ht} &= \frac{1}{\rho A}\int_{\mathcal{I}}\omega^2\,dt,\\
c_{ht} &= \int_{\mathcal{I}}\dot{\lambda}\omega\,dt,\\
d_{ht} &= \int_{\mathcal{I}}\dot{\omega}\lambda\,dt = \nu(T)\lambda(T) - c_{ht},\\
k_{ht} &= EA\int_{\mathcal{I}}\lambda^2\,dt,\\
r_{h\nu}(\nu^*) &= \left(\int_{\mathcal{I}}\dot{q}_{m-1}\omega\,dt\right)\nu^* - \frac{1}{\rho A}\left(\int_{\mathcal{I}}p_{m-1}\omega\,dt\right)\nu^*,\\
r_{h\mu}(\mu^*) &= \left(\int_{\mathcal{I}}f\lambda\,dt\right)\mu^* - \left(\int_{\mathcal{I}}\dot{p}_{m-1}\lambda\,dt\right)\mu^* - EA\left(\int_{\mathcal{I}}\frac{\partial q_{m-1}}{\partial x}\lambda\,dt\right)\mu^{*\prime}.
\end{aligned}
$$

We remark that $c_{ht}$ and $d_{ht}$ are constant while $m_{ht}$ and $k_{ht}$ could possibly depend on the space variable.

To construct the problem in time, we suppose that $\mu$ and $\nu$ are known and look for $\lambda$ and $\omega$. Choosing the tests functions as $q^*(x,t) = \mu(x)\lambda^*(t)$ and $p^*(x,t) = \nu(x)\omega^*(t)$ in (2.36) and (2.37), one obtains:

$$\begin{aligned}
&\int_{\mathcal{I}} c_{hx}\dot{\lambda}\omega^* - m_{hx}\omega\omega^*\,dt = \int_{\mathcal{I}} r_{h\omega}(\omega^*)\,dt, && \forall\omega^*,\\
&\int_{\mathcal{I}} c_{hx}\dot{\omega}\lambda^* + k_{hx}\lambda\lambda^*\,dt = \int_{\mathcal{I}} r_{h\lambda}(\lambda^*)\,dt + \left(\int_{\mathcal{I}} \lambda^* g(t)\,dt\right)\mu(\ell), && \forall\lambda^*,
\end{aligned} \tag{2.39}$$

with:

$$\begin{aligned}
m_{hx} &= \int_\Omega \frac{1}{\rho A}\nu^2\,dx,\\
c_{hx} &= \int_\Omega \mu\nu\,dx,\\
k_{hx} &= \int_\Omega EA(\mu')^2\,dx,\\
r_{h\omega}(\omega^*) &= \left(\int_\Omega \frac{1}{\rho A}p_{m-1}\nu\,dx\right)\omega^* - \left(\int_\Omega \dot{q}_{m-1}\nu\,dx\right)\omega^*,\\
r_{h\lambda}(\lambda^*) &= \left(\int_\Omega f\mu\,dx\right)\lambda^* - \left(\int_\Omega \dot{p}_{m-1}\mu\,dx\right)\lambda^* - \left(\int_\Omega EA\frac{\partial q_{m-1}}{\partial x}\mu'\,dx\right)\lambda^*.
\end{aligned}$$

### Discrete formulation

For the discretization of Problem (2.38) in space, we are looking for finite element solutions $\mu_h \in V^h$ and $\nu_h \in V^h$ of $\mu$ and $\nu$, respectively, such that:

$$\begin{aligned}
\mu(x) \approx \mu_h(x) &= \sum_{j=1}^{n} \mu_j\phi_j(x),\\
\nu(x) \approx \nu_h(x) &= \sum_{j=1}^{n} \nu_j\phi_j(x),
\end{aligned}$$

that satisfy the system of coupled equations:

$$\begin{aligned}
&\int_\Omega m_{ht}\nu_h\phi_i - c_{ht}\mu_h\phi_i\,dx = \int_\Omega r_{h\nu}(\phi_i)\,dx, && \forall i = 1,\dots,n,\\
&\int_\Omega d_{ht}\nu_h\phi_i + k_{ht}\mu_h'\phi_i' dx = \int_\Omega r_{h\mu}(\phi_i)\,dx + \left(\int_{\mathcal{I}} \lambda g(t)\,dt\right)\phi_i(\ell), && \forall i = 1,\dots,n,
\end{aligned}$$

which can be equivalently written as:

$$\begin{bmatrix} K_{ht} & D_{ht} \\ -C_{ht} & M_{ht} \end{bmatrix} \begin{bmatrix} U_{h\mu} \\ U_{h\nu} \end{bmatrix} = \begin{bmatrix} R_{h\mu} \\ R_{h\nu} \end{bmatrix}. \tag{2.40}$$

The stiffness and mass matrices are given here as:

$$\begin{aligned} K_{ht} &= K \left( \int_{\mathcal{I}} \lambda^2 \, dt \right), & D_{ht} &= \bar{M} \left( \int_{\mathcal{I}} \dot{\omega} \lambda \, dt \right), \\ C_{ht} &= \bar{M} \left( \int_{\mathcal{I}} \omega \dot{\lambda} \, dt \right), & M_{ht} &= \bar{\bar{M}} \left( \int_{\mathcal{I}} \omega^2 \, dt \right), \end{aligned}$$

while the vectors $U_{h\mu}$ and $U_{h\nu}$ are the vectors of the degrees of freedom associated with $\mu_h$ and $\nu_h$, respectively, and the loading vectors $R_{h\mu}$ and $R_{h\nu}$ are the residual vectors corresponding to $r_{h\mu}$ and $r_{h\nu}$.

Proceeding as before, the discretization in time of the system of equations (2.39) using the algorithm described in Section 2.2.3 leads to:

$$\begin{bmatrix} h_t k_{hx} & 2c_{hx} \\ 2c_{hx} & -h_t m_{hx} \end{bmatrix} \begin{bmatrix} \lambda^{n+1} \\ \omega^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t k_{hx} & 2c_{hx} \\ 2c_{hx} & h_t m_{hx} \end{bmatrix} \begin{bmatrix} \lambda^{n} \\ \omega^{n} \end{bmatrix} + h_t \begin{bmatrix} r^n_{h\lambda} + r^{n+1}_{h\lambda} \\ r^n_{h\omega} + r^{n+1}_{h\omega} \end{bmatrix}, \quad \forall n = 0, \ldots, n_t - 1. \tag{2.41}$$

We observe that the above system is slightly different from the one obtained by the Lagrangian approach (2.35).

### 2.3.3 Updating procedure of the temporal modes and Gram-Schmidt process

Let us consider a separable function $q(x,t)$. At the $m^{\text{th}}$ enrichment, the PGD approximation of $q$ is given by:

$$q(x,t) \approx q_m(x,t) = q_0(x,t) + \sum_{i=1}^{m} \lambda_i(t)\mu_i(x) = q_{m-1}(x,t) + \lambda_m(t)\mu_m(x).$$

The algorithm used here to construct the modes is based on a greedy approach; namely, each enrichment step aims at the determination of the new mode $(\mu_m, \lambda_m)$. The pair $(\mu_m, \lambda_m)$ is thus computed based on the information contained in $(\mu_k, \lambda_k)_{1 \leqslant k \leqslant m-1}$. However, the previously computed modes $(\mu_k, \lambda_k)_{1 \leqslant k \leqslant m-1}$ do not benefit from the new information introduced by $(\mu_m, \lambda_m)$.

One idea to improve the convergence of the PGD approximation is to update the temporal modes in a global manner [53, 54, 88, 89]. In other words, after a new mode is found, the updating algorithm reevaluates the modes $(\lambda_k)_{1\leqslant k\leqslant m}$ in order to obtain a better combination of the spatial modes $(\mu_k)_{1\leqslant k\leqslant m}$.

This procedure will significantly improve the convergence of the PGD at a fairly low computational cost, which only depends on $n_t$ and $m$ [54]. However, this procedure requires the computation of some matrices that can become ill-conditioned with the increase of the number of modes. If the later occurs, the procedure causes instabilities.

### Lagrangian update

We consider (2.17) and solve for $(\lambda_k)_{1\leqslant k\leqslant m}$, with $(\mu_k)_{1\leqslant k\leqslant m}$ known, using the following trial and test functions:

$$q_m(x,t) = q_0(x,t) + \sum_{i=1}^{m} \lambda_i(t)\mu_i(x), \qquad q^*(x,t) = \sum_{i=1}^{m} \lambda_i^*(t)\mu_i(x).$$

After discretization, we obtain the following system of equations:

$$\begin{bmatrix} h_t K_{\ell x} & 2M_{\ell x} \\ 2\bar{M}_{\ell x} & -h_t \bar{M}_{\ell x} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n+1} \\ \boldsymbol{\omega}^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t K_{\ell x} & 2M_{\ell x} \\ 2\bar{M}_{\ell x} & h_t \bar{M}_{\ell x} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n} \\ \boldsymbol{\omega}^{n} \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{r}_{\ell\lambda}^{n} + \boldsymbol{r}_{\ell\lambda}^{n+1} \\ 0 \end{bmatrix}, \ \forall n = 1, \ldots, n_t - 1, \tag{2.42}$$

where:

$$\begin{aligned} \boldsymbol{\lambda}(t) &= (\lambda_1(t), \lambda_2(t), \ldots, \lambda_m(t)), \\ \boldsymbol{\omega}(t) &= (\dot{\lambda}_1(t), \dot{\lambda}_2(t), \ldots, \dot{\lambda}_m(t)), \\ \boldsymbol{r}_{\ell\lambda}(t) &= (r_{\ell\lambda 1}(t), r_{\ell\lambda 2}(t), \ldots, r_{\ell\lambda m}(t)), \end{aligned}$$

and:

$$\begin{aligned} K_{\ell x} &= \left[U_{\ell\mu_i}^T K U_{\ell\mu_j}\right]_{1\leqslant i,j\leqslant m}, \\ M_{\ell x} &= \left[U_{\ell\mu_i}^T M U_{\ell\mu_j}\right]_{1\leqslant i,j\leqslant m}, \\ \bar{M}_{\ell x} &= \left[U_{\ell\mu_i}^T \bar{M} U_{\ell\mu_j}\right]_{1\leqslant i,j\leqslant m}. \end{aligned}$$

### Hamiltonian update

We consider here the system (2.23) and solve for $(\lambda_k, \omega_k)_{1\leqslant k\leqslant m}$, with $(\mu_k, \nu_k)_{1\leqslant k\leqslant m}$ known, using the following trial and test functions:

$$\begin{aligned} q_m(x,t) &= q_0(x,t) + \sum_{i=1}^{m} \lambda_i(t)\mu_i(x), \qquad & q^*(x,t) &= \sum_{i=1}^{m} \lambda_i^*(t)\mu_i(x), \\ p_m(x,t) &= p_0(x,t) + \sum_{i=1}^{m} \omega_i(t)\nu_i(x), \qquad & p^*(x,t) &= \sum_{i=1}^{m} \omega_i^*(t)\nu_i(x). \end{aligned}$$

Following discretization of the equations, we obtain:

$$\begin{bmatrix} h_t K_{hx} & 2C_{hx} \\ 2C_{hx}^T & -h_t M_{hx} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n+1} \\ \boldsymbol{\omega}^{n+1} \end{bmatrix} = \begin{bmatrix} -h_t K_{hx} & 2C_{hx} \\ 2C_{hx}^T & h_t M_{hx} \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda}^{n} \\ \boldsymbol{\omega}^{n} \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{r}_{h\lambda}^n + \boldsymbol{r}_{h\lambda}^{n+1} \\ \boldsymbol{r}_{h\omega}^n + \boldsymbol{r}_{h\omega}^{n+1} \end{bmatrix}, \ \forall n = 1, \dots, n_t - 1, \tag{2.43}$$

where:

$$\begin{aligned} \boldsymbol{\lambda}(t) &= (\lambda_1(t), \lambda_2(t), \dots, \lambda_m(t)), \\ \boldsymbol{\omega}(t) &= (\omega_1(t), \omega_2(t), \dots, \omega_m(t)), \\ \boldsymbol{r}_{h\omega}(t) &= (r_{h\omega 1}(t), r_{h\omega 2}(t), \dots, r_{h\omega m}(t)), \\ \boldsymbol{r}_{h\lambda}(t) &= (r_{h\lambda 1}(t), r_{h\lambda 2}(t), \dots, r_{h\lambda m}(t)), \end{aligned}$$

and:

$$\begin{aligned} K_{hx} &= \left[U_{h\mu_i}^T K U_{h\mu_j}\right]_{1\leqslant i,j\leqslant m}, \\ M_{hx} &= \left[U_{h\nu_i}^T \bar{\bar{M}} U_{h\nu_j}\right]_{1\leqslant i,j\leqslant m}, \\ C_{hx} &= \left[U_{h\mu_i}^T \bar{M} U_{h\nu_j}\right]_{1\leqslant i,j\leqslant m}. \end{aligned}$$

### Gram-Schmidt process

The question of the metric with respect to which the spatial basis should be orthogonalized in the Gram-Schmidt procedure arises: should one orthogonalize with respect to $K$, $M$, or any other symmetric positive definite matrix? The matrices $K_{\ell x}$, $M_{\ell x}$, $\bar{M}_{\ell x}$, $K_{hx}$, and $M_{hx}$ introduced in the previous section have special properties. They are called Gram matrices. Their coefficients result from scalar products with respect to discrete metrics associated with the matrices $K$, $M$, $\bar{M}$, or $\bar{\bar{M}}$.

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix and let $(u_1, \dots, u_m)$ be a family of vectors of $\mathbb{R}^n$. One can then associate a scalar product with $A$ such that $\langle u_i, u_j \rangle_A = u_i^T A u_j$ and a norm such that $\|u_i\|_A = \sqrt{\langle u_i, u_i \rangle_A}$. Let $G \in \mathbb{R}^{m \times m}$ be such that $G = \left[\langle u_i, u_j \rangle_A\right]_{1 \leqslant i,j \leqslant m}$, the Gram matrix associated with $A$ and $(u_1, \ldots, u_m)$. By virtue of the scalar product properties, $G$ is symmetric positive semi-definite and is invertible if and only if the vectors $(u_1, \ldots, u_m)$ are linearly independent.

The update procedures described above strongly rely on the fact that the computed Gram matrices are well conditioned. Yet, given the properties of the Gram matrices, some choices regarding the metric in the Gram-Schmidt procedure are more suitable than others. In order to ensure that the condition numbers of the matrices are kept small, the Gram-Schmidt procedure is therefore performed as follows:

- For the Lagrangian update: one spatial basis $(U_{\ell \mu_k})_{1 \leqslant k \leqslant m}$ is built and orthogonalized with respect to $K$ and then normalized. In other words, following the Gram-Schmidt procedure, $K_{\ell x}$ should be equal to $I_m$, where $I_m$ is the identity matrix of size $m$. $M_{\ell x}$ and $\bar{M}_{\ell x}$ will not have a particular form but their conditioning numbers should remain low as long as the basis vectors remain linearly independent;

- For the Hamiltonian update: two spatial bases $(U_{h\mu_k})_{1 \leqslant k \leqslant m}$ and $(U_{h\nu_k})_{1 \leqslant k \leqslant m}$ are built for $q$ and $p$, respectively. An optimal choice, which, to the best of our knowledge constitutes a new result, is to orthogonalize $(U_{h\mu_k})_{1 \leqslant k \leqslant m}$ and $(U_{h\nu_k})_{1 \leqslant k \leqslant m}$ with respect to $K$ and $\bar{\bar{M}}$, respectively, and then to normalize the vectors. Following the Gram-Schmidt procedure, $K_{hx} = M_{hx} = I_m$ and their conditioning remains optimal.

### 2.3.4 Adaptive fixed-point algorithm

We briefly describe in this section the fixed-point algorithms for the Lagrangian and Hamiltonian formulations of the PGD approach. In particular, we propose in the case of the Hamiltonian formulation an algorithm that allows one to accelerate the convergence toward the enrichment modes associated with the generalized coordinates and the conjugate fields.

#### Lagrangian fixed-point iteration

For the sake of simplicity in the notation, we will simply use $\mu$ and $\lambda$ to refer here to the finite element solution $\mu_h$ (or the vector of degrees of freedom $U_{\ell\mu}$) and discrete solution $(\lambda^0, \lambda^1, \ldots, \lambda^{n_t})$. Moreover, we introduce:

- $\mathcal{S}_\ell : \lambda \mapsto \mu$, the operator that solves the system (2.34) for $\mu$ with $\lambda$ given;

- $\mathcal{T}_\ell : \mu \mapsto \lambda$, the operator that solves the system (2.35) for $\lambda$ with $\mu$ given.

Let $j_{\max}$ and $\epsilon$ denote the user-defined maximum number of iterations and tolerance. The fixed-point algorithm for the Lagrangian approach is described as a pseudocode in Algorithm 1. Note that the norm subscripted with $L^2$ is defined as follows for square-integrable functions:

$$\|f\|_{L^2}^2 = \int_{\mathcal{I}} \int_{\Omega} f(x,t)^2 \; dxdt$$

**Algorithm 1** fixed-point algorithm for the Lagrangian formulation

```
**Initialize** $\lambda_0$ and $\mu_0$, $j \leftarrow 0$, $s \leftarrow \epsilon + 1$
**while** $j < j_{\max}$ and $s > \epsilon$ **do**
    Increment the iteration counter: $j \leftarrow j + 1$
    Compute new spatial mode: $\mu_j \leftarrow \mathcal{S}_\ell(\lambda_{j-1})$
    Normalize: $\mu_j \leftarrow \mu_j / \|\mu_j\|_K$
    Compute new temporal mode: $\lambda_j \leftarrow \mathcal{T}_\ell(\mu_j)$
    Calculate the PGD difference $\Delta \leftarrow \mu_j\lambda_j - \mu_{j-1}\lambda_{j-1}$
                    and average $\Sigma \leftarrow \frac{1}{2}\left(\mu_j\lambda_j + \mu_{j-1}\lambda_{j-1}\right)$
    Evaluate the stagnation coefficient: $s \leftarrow \|\Delta\|_{L^2} / \|\Sigma\|_{L^2}$
**end while**
**Return** the modes $\lambda \leftarrow \lambda_j$ and $\mu \leftarrow \mu_j$
```

### Hamiltonian fixed-point iteration

As before, the notations $\mu$, $\nu$, $\lambda$, $\omega$ will be used to refer to the finite element solutions $\mu_h$ and $\nu_h$ and the discrete solutions $(\lambda^0, \lambda^1, \ldots, \lambda^{n_t})$ and $(\omega^0, \omega^1, \ldots, \omega^{n_t})$. Moreover, we consider the operators:

- $\mathcal{S}_h : (\lambda, \omega) \mapsto (\mu, \nu)$, the operator that solves the system (2.40) for $(\mu, \nu)$ with $(\lambda, \omega)$ given;
- $\mathcal{T}_h : (\mu, \nu) \mapsto (\lambda, \omega)$, the operator that solves the system (2.41) for $(\lambda, \omega)$ with $(\mu, \nu)$ given.

We have observed in the numerical experiments that convergence of the pairs $(\mu, \lambda)$ and $(\nu, \omega)$ does not necessarily happens at the same time. We therefore propose to decouple the iterative process as follows:

- If convergence is reached on the mode $\mu\lambda$, we fix the values of $\mu$ and $\lambda$ and then solve (2.44) and (2.45) for $\nu$ and $\omega$ until convergence is reached:

$$M_{ht}\nu = C_{ht}\mu + R_{h\nu} \tag{2.44}$$

$$h_t m_{hx}\omega^{n+1} = -h_t m_{hx}\omega^n + 2c_{hx}\left(\lambda^{n+1} - \lambda^n\right) + h_t\left(r_{h\omega}^n + r_{h\omega}^{n+1}\right), \quad \forall n = 0, \ldots, n_t - 1. \tag{2.45}$$

  The operators associated with (2.44) and (2.45) will be denoted by $\mathcal{S}_{h\nu} : \omega \mapsto \nu$, $\mathcal{T}_{h\omega} : \nu \mapsto \omega$, respectively.

- If convergence is reached on the mode $\nu\omega$, we fix the values of $\nu$ and $\omega$ and then solve (2.46) and (2.47) for $\mu$ and $\lambda$ until convergence is reached:

$$K_{ht}\mu = -D_{ht}\nu + R_{h\mu} \tag{2.46}$$

$$h_t k_{hx}\lambda^{n+1} = -h_t k_{hx}\lambda^n + 2c_{hx}\left(\omega^n - \omega^{n+1}\right) + h_t\left(r_{h\lambda}^n + r_{h\lambda}^{n+1}\right), \quad \forall n = 0, \ldots, n_t - 1. \tag{2.47}$$

  The operators associated with (2.46) and (2.47) will be denoted by $\mathcal{S}_{h\mu} : \lambda \mapsto \mu$ and $\mathcal{T}_{h\lambda} : \mu \mapsto \lambda$, respectively.

The fixed-point algorithm for the Hamiltonian approach is detailed as a pseudocode in Algorithm 2. It is essential to notice that the dimensions of the systems to solve with the H-PGD are twice as big as the ones with the L-PGD. The advantage of this implementation is twofold. On the one hand, it allows to control the convergence of the fields separately. On the other hand, this fixed-point algorithm stops iterating on the field that has converged and therefore iterates on systems twice as small, i.e. of the same dimension as the L-PGD.

## 2.4 Numerical results and discussion

### 2.4.1 Test cases

The objective of this section is to present several numerical examples in order to compare the PGD solutions obtained using the Lagrangian formulation and the Hamiltonian formulation. We shall consider in all experiments a one-dimensional bar of length $\ell = 0.2$ m with material properties $E = 220$ GPa, $\rho = 7000$ kg/m$^3$, and $A = 10^{-3}$ m$^2$. Unless stated otherwise, we will solve the differential equation (2.1) with $f(x,t) = 0$, $\forall x \in \Omega$, $\forall t \in \mathcal{I}$, with $T = 1.15$ ms. Moreover, we will assume that the bar is always fixed at $x = 0$, i.e. $u(0,t) = 0$, $\forall t \in \mathcal{I}$. We shall consider five scenarios, that may differ one from the other by the choice of initial

**Algorithm 2** fixed-point algorithm for the Hamiltonian formulation

```
**Initialize** $\lambda_0$, $\omega_0$, $\mu_0$, and $\nu_0$, $j \leftarrow 0$, $s_q \leftarrow \epsilon + 1$, $s_p \leftarrow \epsilon + 1$
**while** $j < j_{\max}$ and $(s_q > \epsilon$ or $s_p > \epsilon)$ **do**
    Increment the iteration counter: $j \leftarrow j + 1$
    **if** $s_q < \epsilon$ **then**
        Compute new spatial mode: $\nu_j \leftarrow \mathcal{S}_{h\nu}(\omega_{j-1})$
        Normalize: $\nu_j \leftarrow \nu_j / \|\nu_j\|_M$
        Compute new temporal mode: $\omega_j \leftarrow \mathcal{T}_{h\omega}(\nu_j)$
        Calculate the PGD difference $\Delta_p \leftarrow \nu_j \omega_j - \nu_{j-1}\omega_{j-1}$
        and average $\Sigma_p \leftarrow \frac{1}{2}\left(\nu_j \omega_j + \nu_{j-1}\omega_{j-1}\right)$
        Evaluate the stagnation coefficient: $s_p \leftarrow \|\Delta_p\|_{L^2} / \|\Sigma_p\|_{L^2}$
        Update $\mu_j \leftarrow \mu_{j-1}$ and $\lambda_j \leftarrow \lambda_{j-1}$ (fixed modes)
    **else if** $s_p < \epsilon$ **then**
        Compute new spatial mode: $\mu_j \leftarrow \mathcal{S}_{h\mu}(\lambda_{j-1})$
        Normalize: $\mu_j \leftarrow \mu_j / \|\mu_j\|_K$
        Compute new temporal mode: $\lambda_j \leftarrow \mathcal{T}_{h\lambda}(\mu_j)$
        Calculate the PGD difference $\Delta_q \leftarrow \mu_j \lambda_j - \mu_{j-1}\lambda_{j-1}$
        and average $\Sigma_q \leftarrow \frac{1}{2}\left(\mu_j \lambda_j + \mu_{j-1}\lambda_{j-1}\right)$
        Evaluate the stagnation coefficient: $s_q \leftarrow \|\Delta_q\|_{L^2} / \|\Sigma_q\|_{L^2}$
        Update $\nu_j \leftarrow \nu_{j-1}$ and $\omega_j \leftarrow \omega_{j-1}$ (fixed modes)
    **else**
        Calculate new spatial modes: $(\mu_j, \nu_j) \leftarrow \mathcal{S}_h(\lambda_{j-1}, \omega_{j-1})$
        Normalize: $\mu_j \leftarrow \mu_j / \|\mu_j\|_K$ and $\nu_j \leftarrow \nu_j / \|\nu_j\|_M$
        Compute new temporal modes: $(\lambda_j, \omega_j) \leftarrow \mathcal{T}_h(\mu_j, \nu_j)$
        Calculate the PGD differences $\Delta_q \leftarrow \mu_j \lambda_j - \mu_{j-1}\lambda_{j-1}$
                                         and $\Delta_p \leftarrow \nu_j \omega_j - \nu_{j-1}\omega_{j-1}$
        Calculate the PGD averages $\Sigma_q \leftarrow \frac{1}{2}\left(\mu_j \lambda_j + \mu_{j-1}\lambda_{j-1}\right)$
                                       and $\Sigma_p \leftarrow \frac{1}{2}\left(\nu_j \omega_j + \nu_{j-1}\omega_{j-1}\right)$
        Evaluate the stagnation coefficient: $s_q \leftarrow \|\Delta_q\|_{L^2} / \|\Sigma_q\|_{L^2}$
                                              and $s_p \leftarrow \|\Delta_p\|_{L^2} / \|\Sigma_p\|_{L^2}$
    **end if**
**end while**
**Return** the modes $\lambda = \lambda_j$, $\omega = \omega_j$, $\mu = \mu_j$, and $\nu = \nu_j$
```

conditions $u_0(x)$ and $v_0(x)$ or the type of boundary condition at the endpoint $x = \ell$:

1. In the first case, we will consider homogeneous initial displacements and velocities, that is $u_0(x) = 0$ and $v_0(x) = 0$, $\forall t \in \mathcal{I}$, and the Neumann condition (2.5) at $x = \ell$ where $g(t)$ is oscillating for $t \leqslant T/2$ and vanishes for $t > T/2$. In these experiments, the PGD solutions will be computed without performing the updating procedure described in Section 2.3.3;
2. In this case, we will repeat the same experiment as above but using the updating procedure of Section (2.3.3) for the calculations of the PGD solutions;
3. In the third case, we keep the homogeneous initial conditions and replace the Neumann condition at $x = \ell$ by an oscillating Dirichlet condition;
4. This experiment will simulate the problem presented in (2.6) for which one has the analytical solution (2.9); however, we will restrict the time interval to $T = 0.14$ ms in order to avoid the spurious oscillations that appear due to the discontinuity in the solution [79, 90];
5. The last case will consider the exact same scenario as in Case 2, but for the presence of an extra linear damping term in the wave equation.

In the following, the PGD solutions obtained from the Lagrangian formalism and the Hamiltonian formalism will be referred to as “L-PGD” and “H-PGD”, respectively. Moreover, we consider two versions of the Lagrangian formulation: “L-PGD1” uses the Crank-Nicolson scheme (also called the implicit trapezoidal rule) for time integration, as presented in the chapter, while “L-PGD2” replaces the Crank-Nicolson scheme by the Newmark method with $\gamma = 1/2$ and $\beta = 1/4$ [15, 83]. The solutions in space will be approximated in terms of continuous piecewise linear polynomial functions for a total of $n = 224$ degrees of freedom, i.e. the domain $\Omega$ is decomposed into $n = 224$ elements of equal size. Likewise, the time interval $\mathcal{I}$ is divided into $n_t = 1025$ sub-intervals of equal size (except in Case 4 where we take $n_t = 1300$). Those values were chosen so that the discretization errors in space and in time are kept small with respect to the truncation errors from the PGD formulation.

### 2.4.2 Comparison method and performance criteria

In order to assess the accuracy of the PGD solutions, we will use as reference solutions, the finite element solutions that are described in Sections 2.2.2 or 2.2.3 and obtained using the same discretization parameters $n = 224$ and $n_t = 1025$. Given a field $u = u(x,t)$ defined on $\Omega \times \mathcal{I}$, we denote by $\epsilon_u$ the relative error in the $L^2$ norm:

$$\epsilon_u = \frac{\|u_m - u_{ref}\|_{L^2}}{\|u_{ref}\|_{L^2}}$$

where $u_m$ is the PGD approximation of rank $m$ of $u$ and $u_{\text{ref}}$ is a very accurate reference solution.

We will study the evolution of the errors with respect to the number of modes $m$ in the PGD solutions and compare these to the errors that one obtains by performing *a posteriori* a Singular Value Decomposition on the reference solutions, except in Case 4 for which we will directly compare the PGD solutions to the analytical solution of problem (2.9). We will in particular look at the error in the energy of the bar over time as the energy (Hamiltonian) in the discrete PGD solution is supposed to remain constant when the external loading vanishes.

Furthermore, we will study the condition numbers of the Gram matrices computed during the temporal update procedure. Condition numbers of such matrices indirectly indicate how well the Gram-Schmidt procedure performs. As soon as the linear independence of the spatial basis is compromised, the procedure does not perform as well and condition numbers may significantly increase. Indeed, the vectors of the spatial basis are linearly independent if and only if the Gram matrices are invertible. More particularly, Gram matrices should be equal to the identity matrix (since the basis vectors are orthonormalized here). Thus, after the 1st enrichment ($m = 1$), the condition numbers are equal to unity. Then, when additional enrichments are considered, two scenarios may occur:

1. The Gram-Schmidt procedure performs well, the Gram matrices remain equal to $I_m$ (the identity matrix of size $m$), and the condition numbers remain equal to unity;

2. Linear independence is compromised and not only some of the off-diagonal coefficients of the Gram matrices become non-null but the Gram matrices are no longer invertible. As a result the condition numbers drastically increase.

### 2.4.3 Case 1: Neumann BC without updating procedure

We approximate in this case Problem (2.1)-(2.5) with $u_0 = 0$, $v_0 = 0$, and the Neumann boundary condition:

$$EA\frac{\partial u}{\partial x}(\ell,t) = g(t) = \begin{cases} F_0\left(1-\cos\left(\omega t\right)\right), & \forall t \in (0, T/2], \\ 0, & \forall t \in (T/2, T). \end{cases}$$

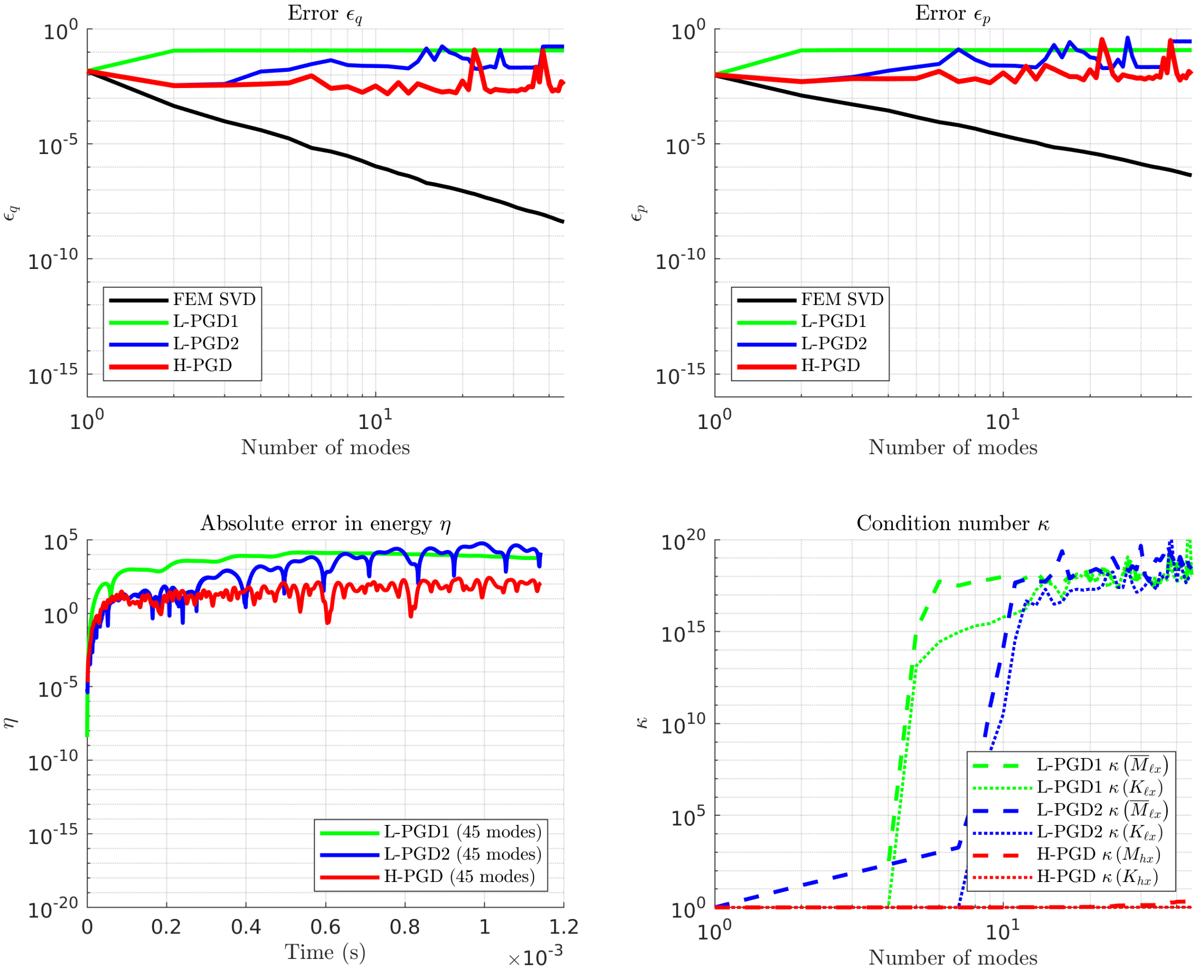


Figure 2.1 Case 1. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3.

where $F_0 = 10^6$ N and $\omega = 4.4 \times 10^4$ rad/s.

We observe in Figure 2.1 that the $L^2$ errors in the generalized coordinates and momenta of the PGD solutions barely decrease, if at all, and that their evolution is non monotonic. However, the H-PGD solution seems to behave slightly better than the L-PGD solutions, especially in terms of the absolute error in energy that remains smaller. We also observe that the condition numbers $\kappa(K_{\ell x})$ and $\kappa(\bar{M}_{\ell x})$ associated with the matrices $K_{\ell x}$ and $\bar{M}_{\ell x}$ increase as soon as the 5th for L-PGD1 and as soon as the 8th enrichment for L-PGD2. This indicates that the Gram-Schmidt algorithm fails to orthogonalize the spatial basis. This is not due to numerical instability and it therefore cannot be corrected by the modified Gram-Schmidt

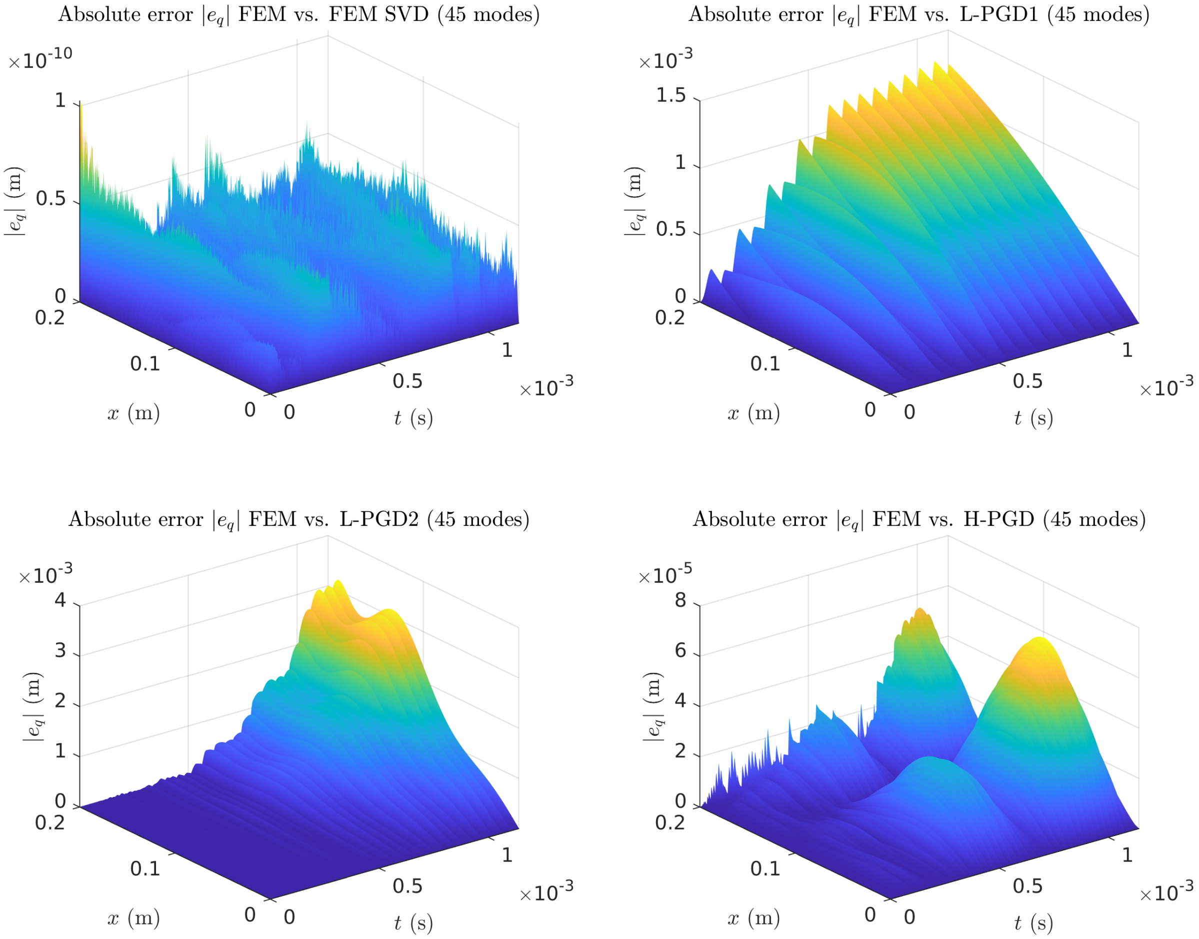


Figure 2.2 Case 1. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field.

algorithm. The increase in the condition number is a consequence of the degeneration of the basis associated with the spatial modes: the added modes compromise the linear independence of the modes. In other words, the new modes do not provide any new information that was not already contained in the previous decomposition.

Figure 2.2 shows the distribution of the errors in space and time for the PGD solutions while Figure 2.3 illustrates the evolution of the displacement fields over time and of the total energy for the FEM solution and the different PGD solutions.

### 2.4.4 Case 2: Neumann BC with updating procedure

We repeat here the same experiment of Case 1 using this time the updating procedure of Section (2.3.3) for the calculations of the PGD solutions. We show in Figure 2.4 the errors in $L^2$ norm and energy and the condition numbers of the matrices introduced in

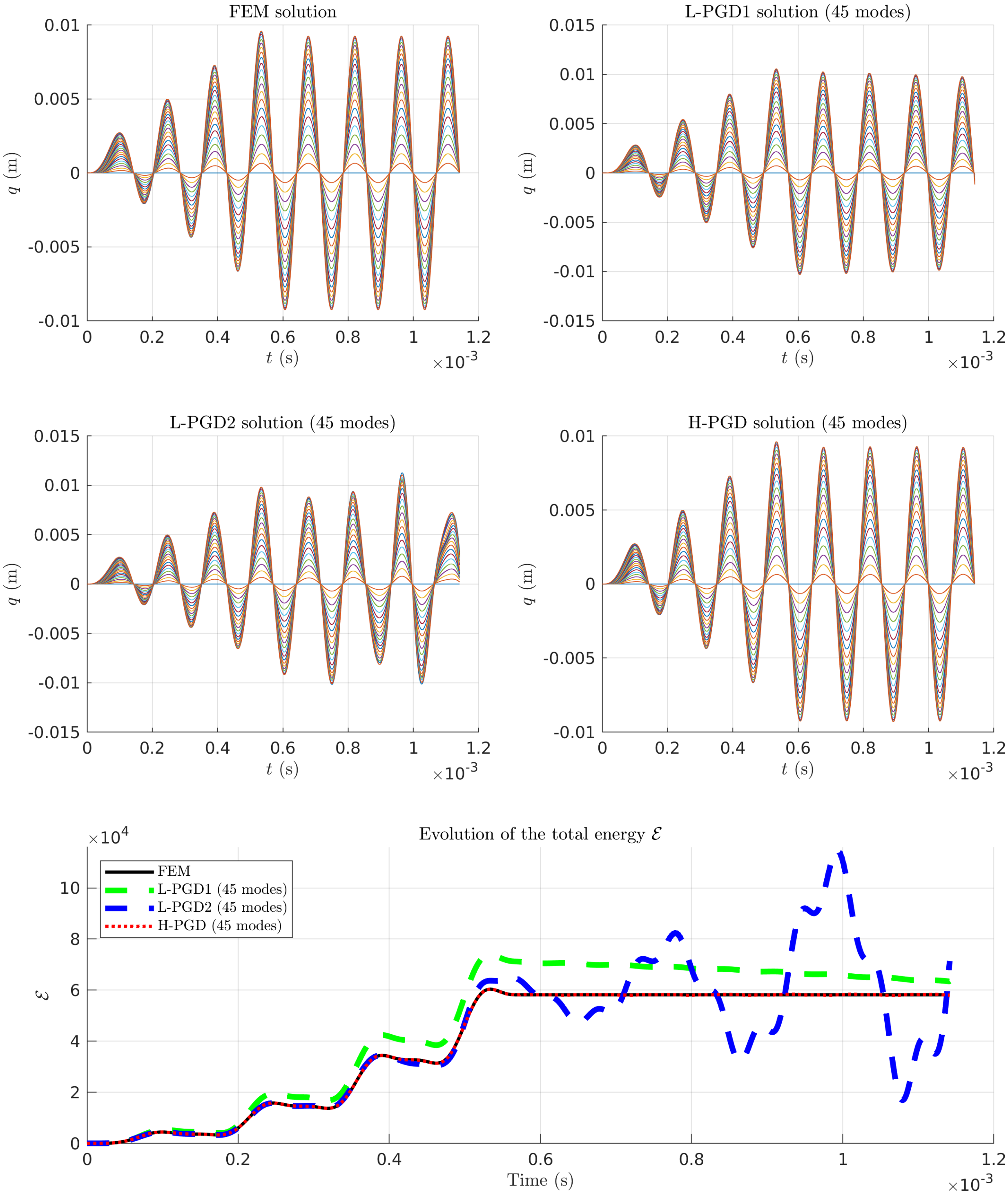


Figure 2.3 Case 1. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time.

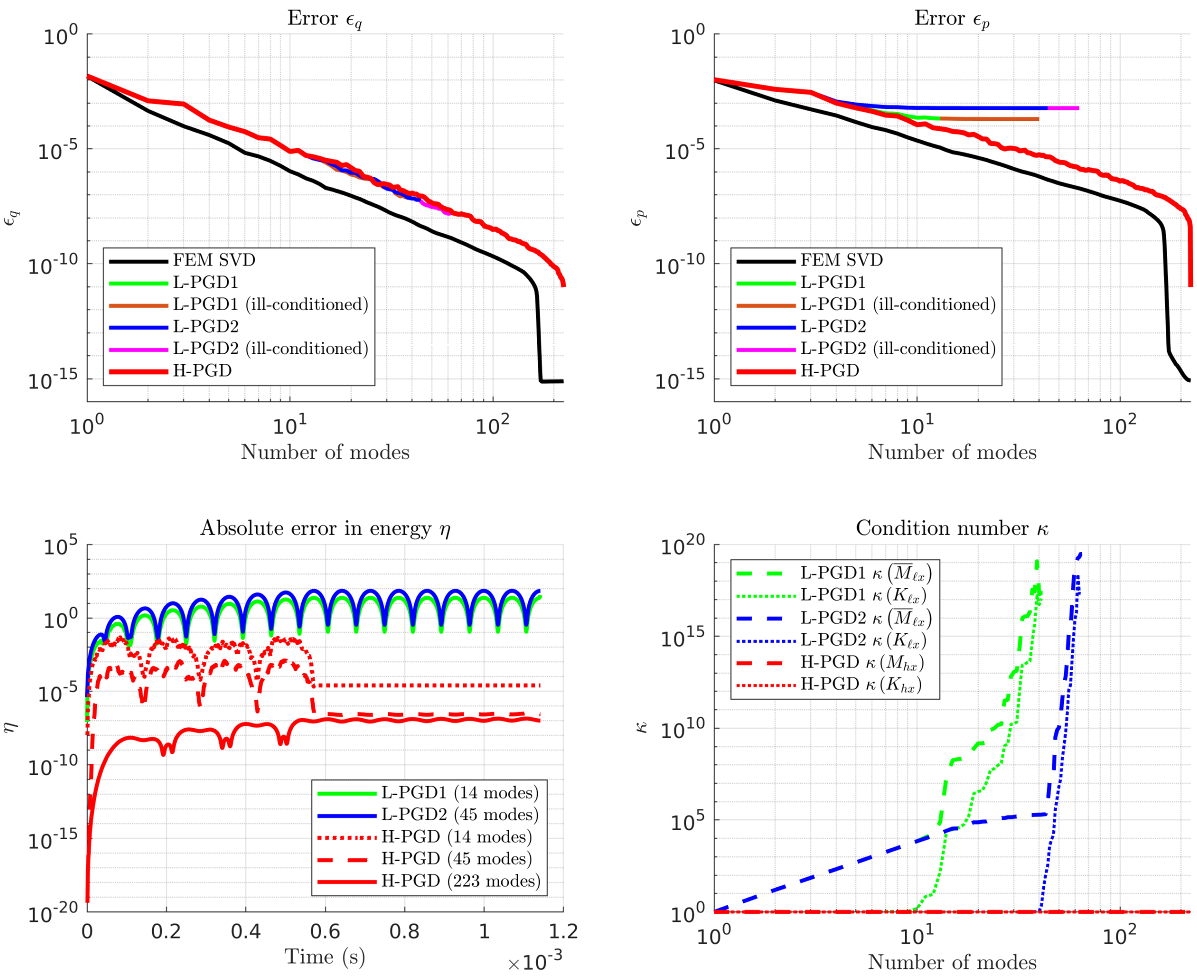


Figure 2.4 Case 2. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3.

Section 2.3.3. We first point out that the updating procedure significantly improves the convergence. Nevertheless, we observe that in the case of the Lagrangian PGD solutions, the matrices for L-PGD1 and L-PGD2 become ill-conditioned as soon as the 14 and 45 modes are reached, respectively. For the L-PGD1 and L-PGD2, the space modes $\mu_k$ are orthogonalized and normalized with respect to Matrix $K$. Thus, the condition number of $K_{\ell x}$ remains low for a dozen of modes (as long as $K_{\ell x} = I_m$) while the condition number of $\bar{M}_{\ell x}$ increases from the beginning. It follows that the condition numbers diverge for the L-PGD1 and L-PGD2 around 40 and 60 modes, respectively. On the other hand, it is remarkable that the matrices in the case of the H-PGD solution always remain well-conditioned. This is explained by the fact that at each enrichment step, H-PGD manages to compute a new mode whose information is

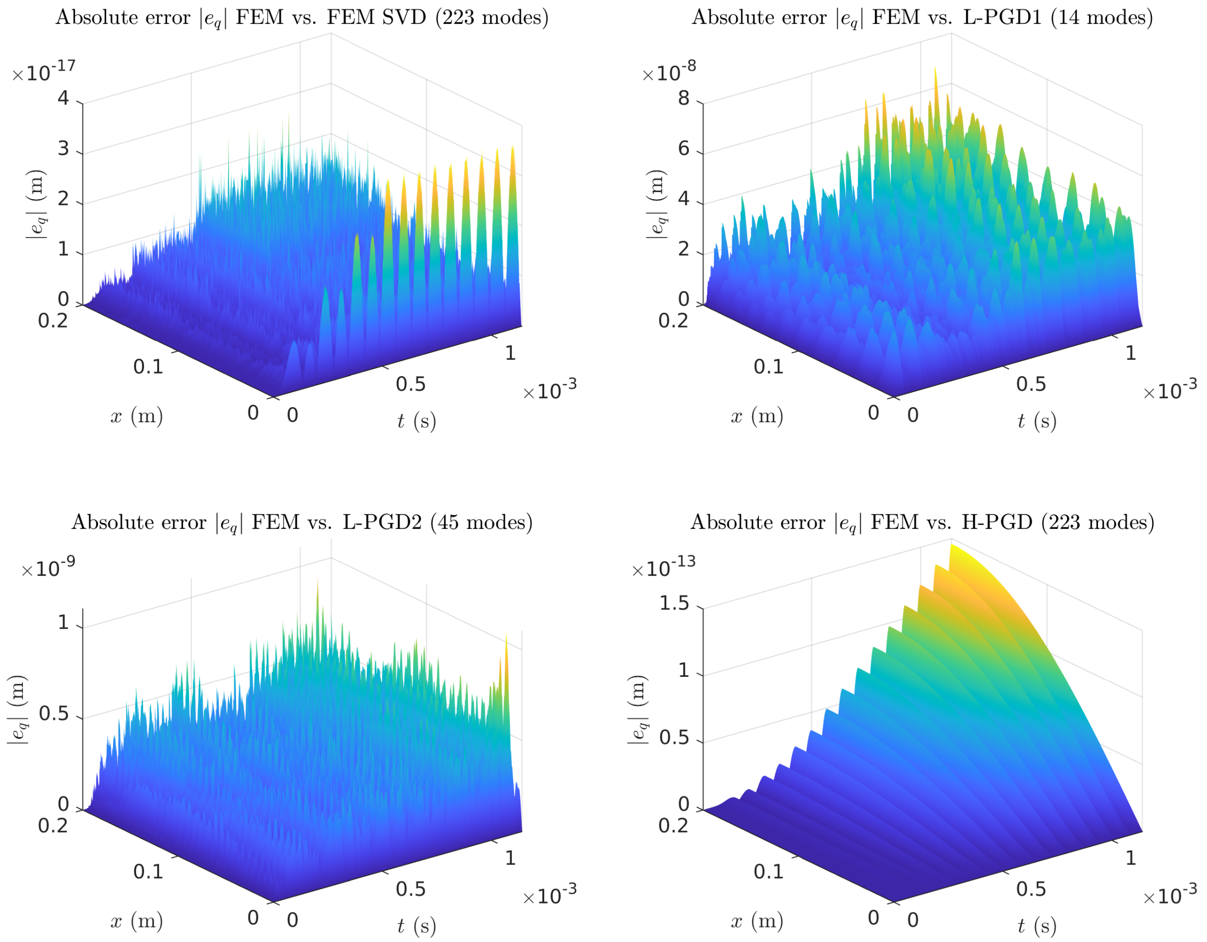


Figure 2.5 Case 2. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field.

not already contained in the old spatial modes. In other words, the Gram-Schmidt algorithm manages to enforce $M_{hx} = I_m$ and $K_{hx} = I_m$.

Before divergence of the L-PGD solutions occurs, we observe that the errors $\epsilon_q$ in the $L^2$ norm in the displacement field follow the same behavior for the three PGDs. However, for L-PGD, the errors in the conjugate momenta $\epsilon_p$ quickly reach a plateau after the calculation of the first dozen modes due to the fact that the matrices become ill-conditioned. It follows that the L-PGD approach fails to identify the relevant modes for $p$. For the H-PGD approach, we see that $\epsilon_p$ keeps decreasing since the method is explicitly designed to compute separate decompositions for both $q$ and $p$. It also implies that the total energy of the system is well approximated for the 223 modes of the H-PGD solution unlike in the case of the L-PGD solutions. We actually observe in Figure 2.5 that the distribution of the errors in space and time for the H-PGD solution remain a few orders of magnitude lower than for the L-PGD

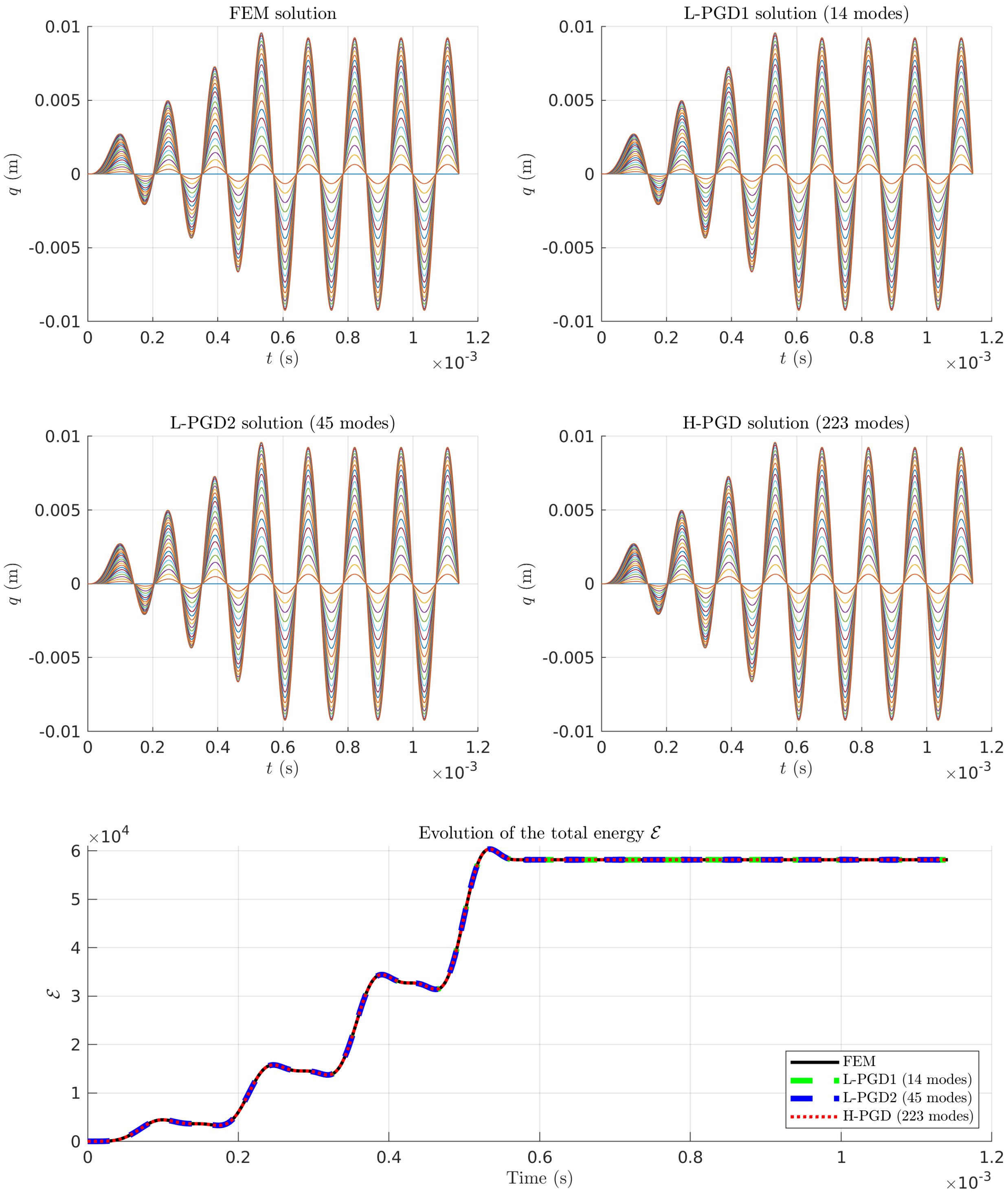


Figure 2.6 Case 2. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time.

solutions, even after the calculation of the 223 modes.

Finally, we show in Figure 2.6 the evolution of the displacement fields over time and of the total energy for the FEM solution and the different PGD solutions. We note that we use in these plots only the first 14 modes for L-PGD1, 45 modes for L-PGD2, and the total of 223 modes for H-PGD. We see that the energy of the system increases as long as the force applied at the end of the beam is non-zero and remains constant once the end of the beam becomes free, as expected. In other words, the Hamiltonian of the system, i.e. the total energy is preserved when the system is conservative.

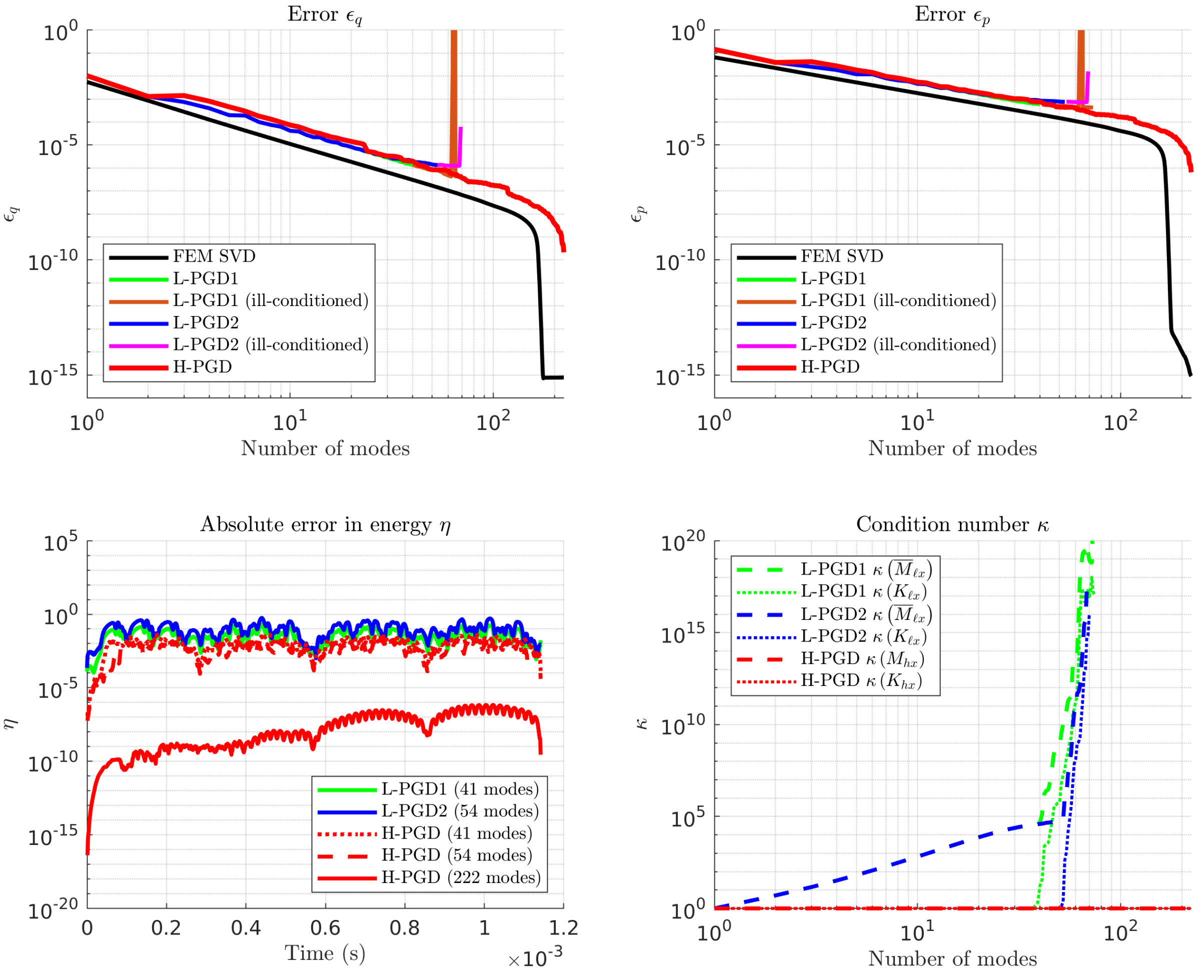
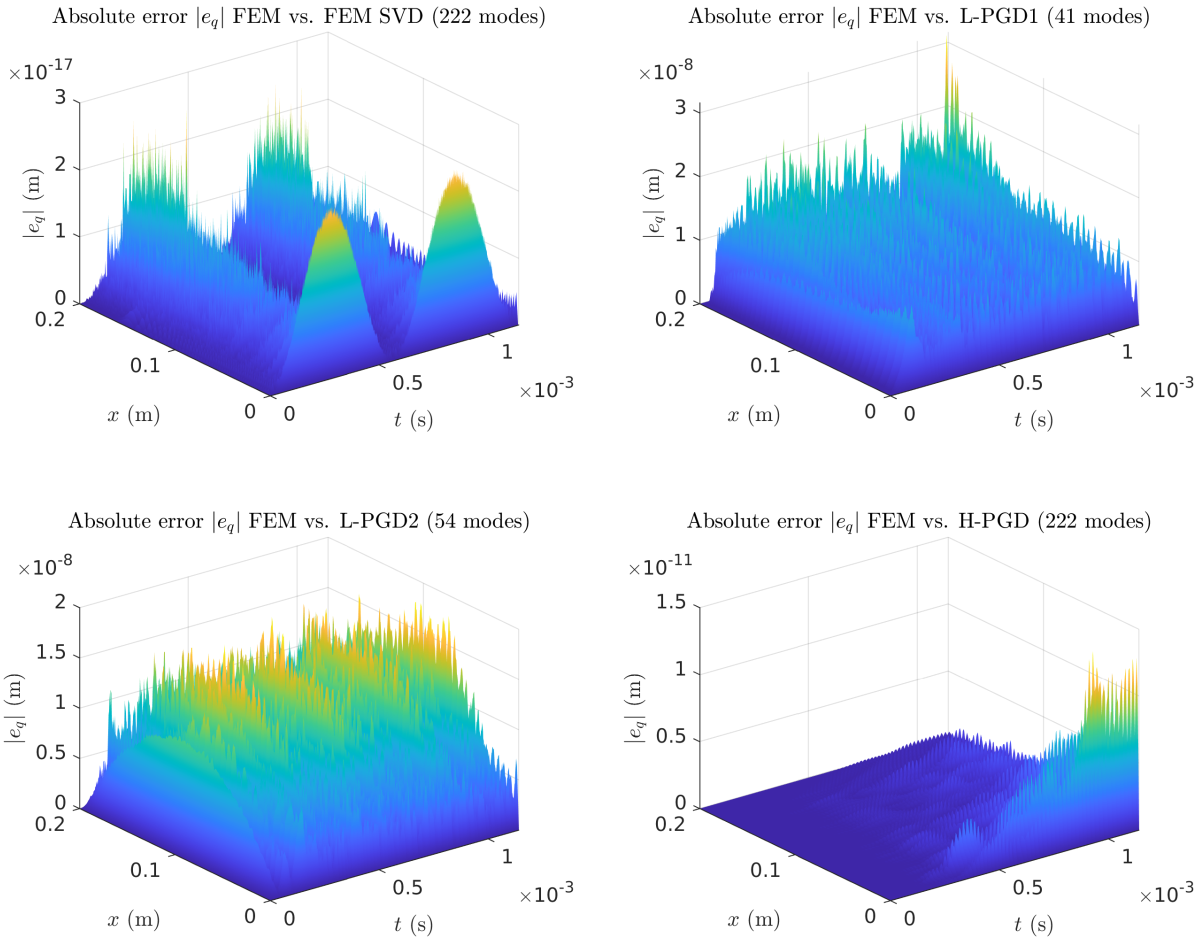


Figure 2.7 Case 3. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3.

### 2.4.5 Case 3: Oscillating Dirichlet BC


Absolute error $|e_q|$ FEM vs. FEM SVD (222 modes)

Absolute error $|e_q|$ FEM vs. L-PGD1 (41 modes)

Absolute error $|e_q|$ FEM vs. L-PGD2 (54 modes)

Absolute error $|e_q|$ FEM vs. H-PGD (222 modes)

Figure 2.8 Case 3. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field.

In this section, we replace the Neumann boundary condition at the end point $x = \ell$ in the previous problem by the oscillatory Dirichlet boundary condition:

$$u(\ell, t) = U_0 \left(1 - \cos\left(\omega t\right)\right), \qquad \forall t \in \mathcal{I},$$

where $U_0 = 5$ mm and $\omega = 1.1 \times 10^4$ rad/s.

We collect the numerical results in Figures 2.7, 2.8 and 2.9. We essentially observe the same behaviors as in the previous test case, except that the matrices associated with the Lagrangian approaches become ill-conditioned after a larger number of computed modes than before and that the relative errors $\epsilon_q$ in the displacement and $\epsilon_p$ in the conjugate momenta quickly diverge rather than reaching a plateau. It is also clear from these results that the

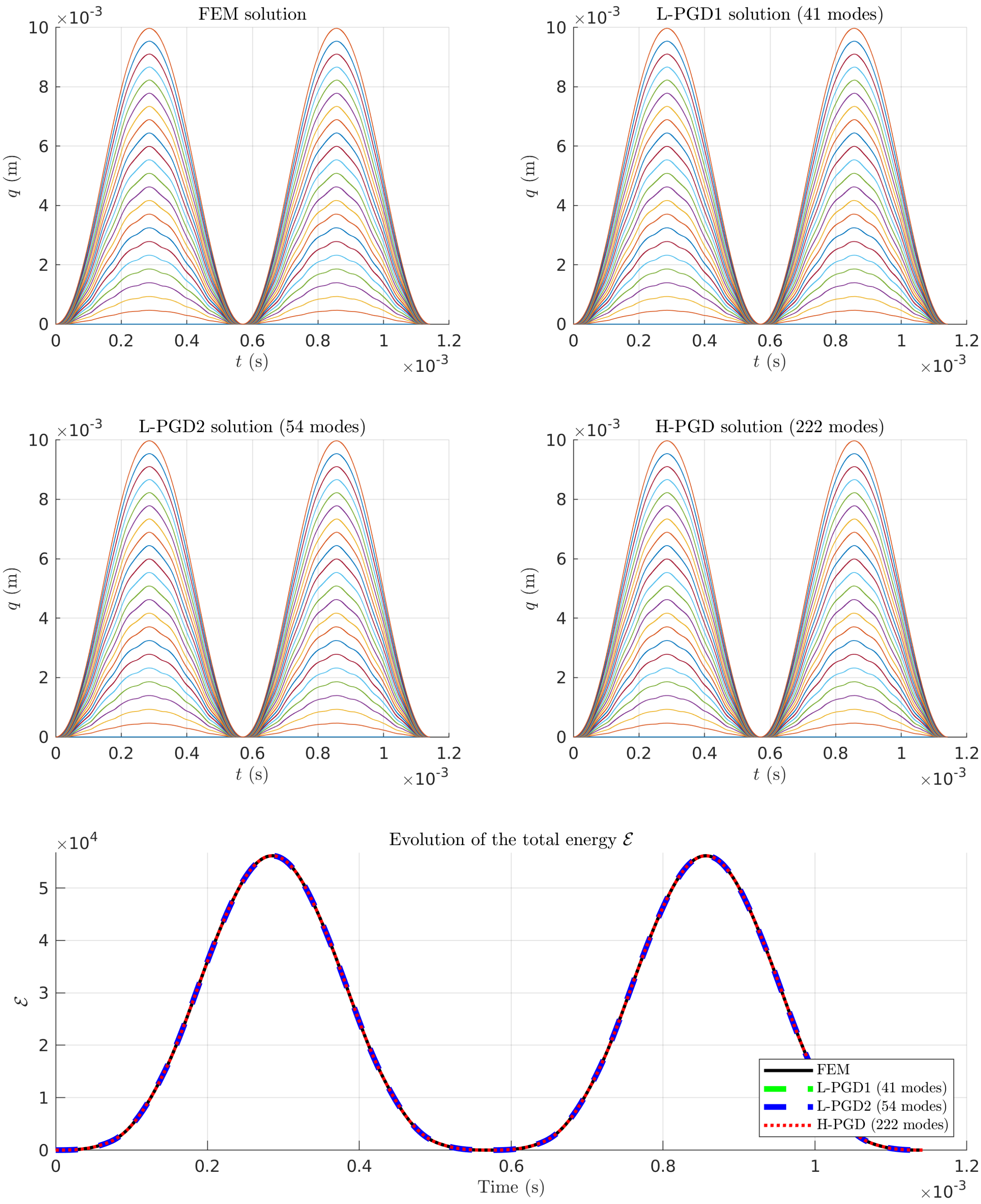


Figure 2.9 Case 3. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time.

H-PGD formulation produces superior results in terms of convergence and accuracy.

### 2.4.6 Case 4: Comparison with analytical solution

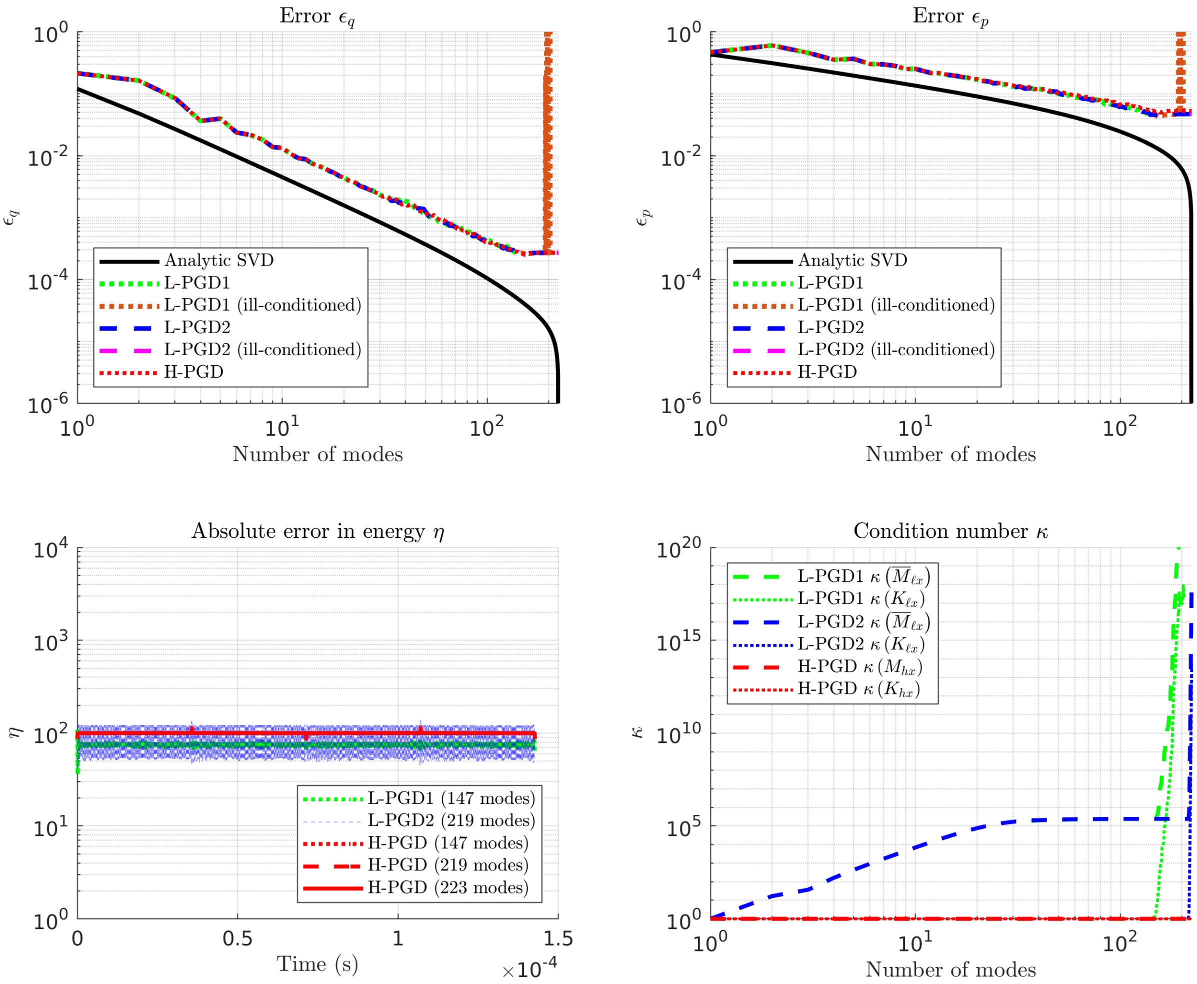
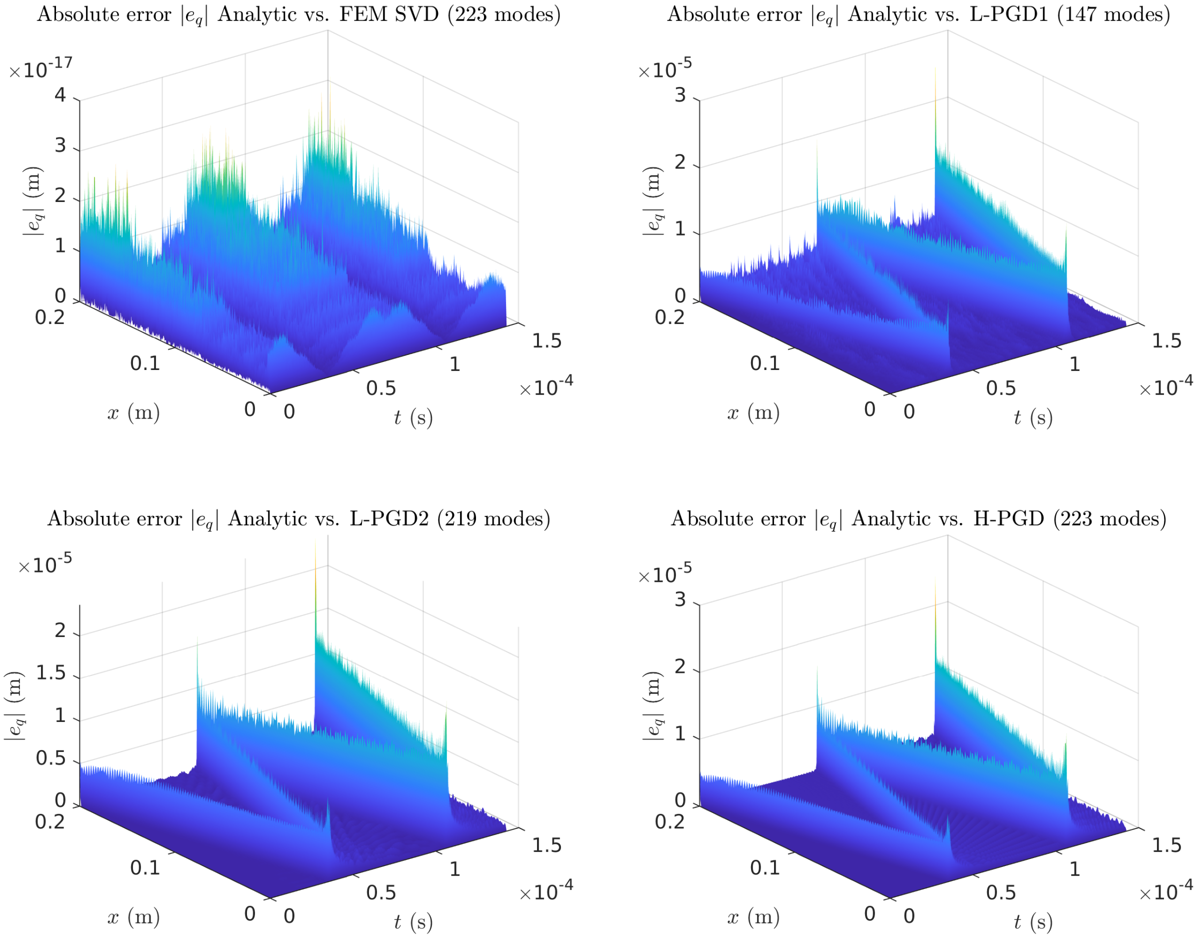


Figure 2.10 Case 4. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3.

We solve in this section the problem formulated in (2.6) where the initial displacement is given by $u_0(x) = Fx/(EA)$, $\forall x \in \Omega$, with $F/(EA) = 0.05$. This test case describes a shock-type wave featuring a discontinuity in the first derivative, see Figure 2.12.

Errors and condition numbers for this test case are shown in Figure 2.10. We observe that the matrices for the L-PGD approaches eventually become ill-conditioned again. Nevertheless, the errors for the three PGD seem to decrease at the same rate. In the analytical solution



Absolute error $|e_q|$ Analytic vs. FEM SVD (223 modes)

Absolute error $|e_q|$ Analytic vs. L-PGD1 (147 modes)

Absolute error $|e_q|$ Analytic vs. L-PGD2 (219 modes)

Absolute error $|e_q|$ Analytic vs. H-PGD (223 modes)

Figure 2.11 Case 4. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field.

provided in (2.9), the error in the truncated displacement retaining only the first $m$ modes is of the order $\mathcal{O}(1/m^2)$. In comparison, the error in the computed PGDs seems to be approximately of the order $\mathcal{O}(1/m^{\frac{3}{2}})$.

We show in Figure 2.11 the distribution of the absolute errors in space and time for the FEM SVD solution, the L-PGD1 and L-PGD2 solutions, and the H-PGD solution. We observe on the one hand that the errors in the FEM SVD solution are negligible. On the other hand, the errors in the last three solutions are of the same order and locally concentrated around the wavefront. These errors are essentially due to the presence of the discontinuity in the solution, which make them difficult to capture. This is out of the scope of this study but we mention that the use of a time-discontinuous Galerkin (TGD) integration scheme could possibly address this issue [49,79]. As a last remark, it seems that the H-PGD solution is less polluted by the large errors than the L-PGD solutions, especially away from the location of

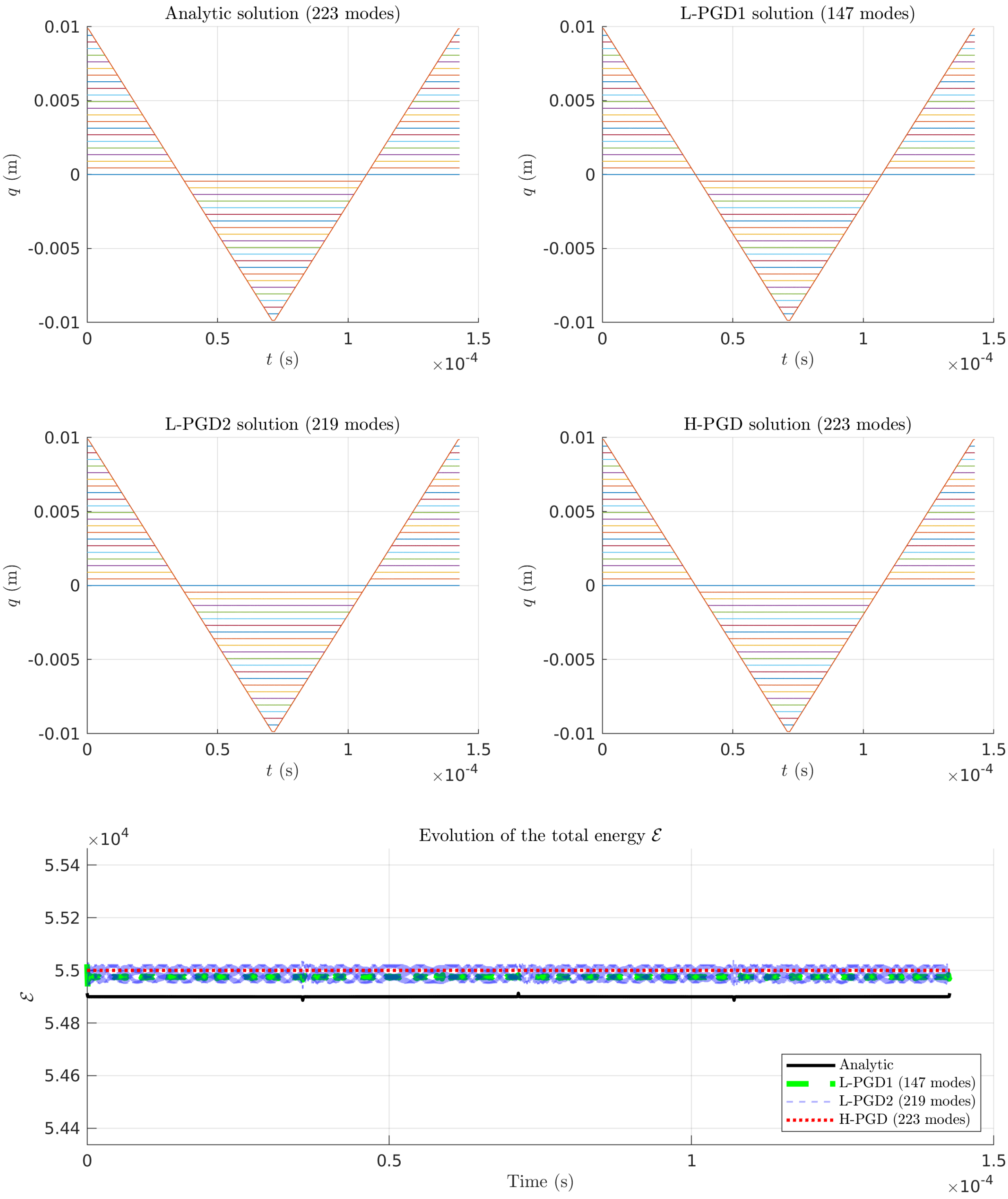


Figure 2.12 Case 4. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time.

the discontinuities and in the vicinity of the end point $x = \ell$.

### 2.4.7 Case 5: Damped bar with Neumann BC

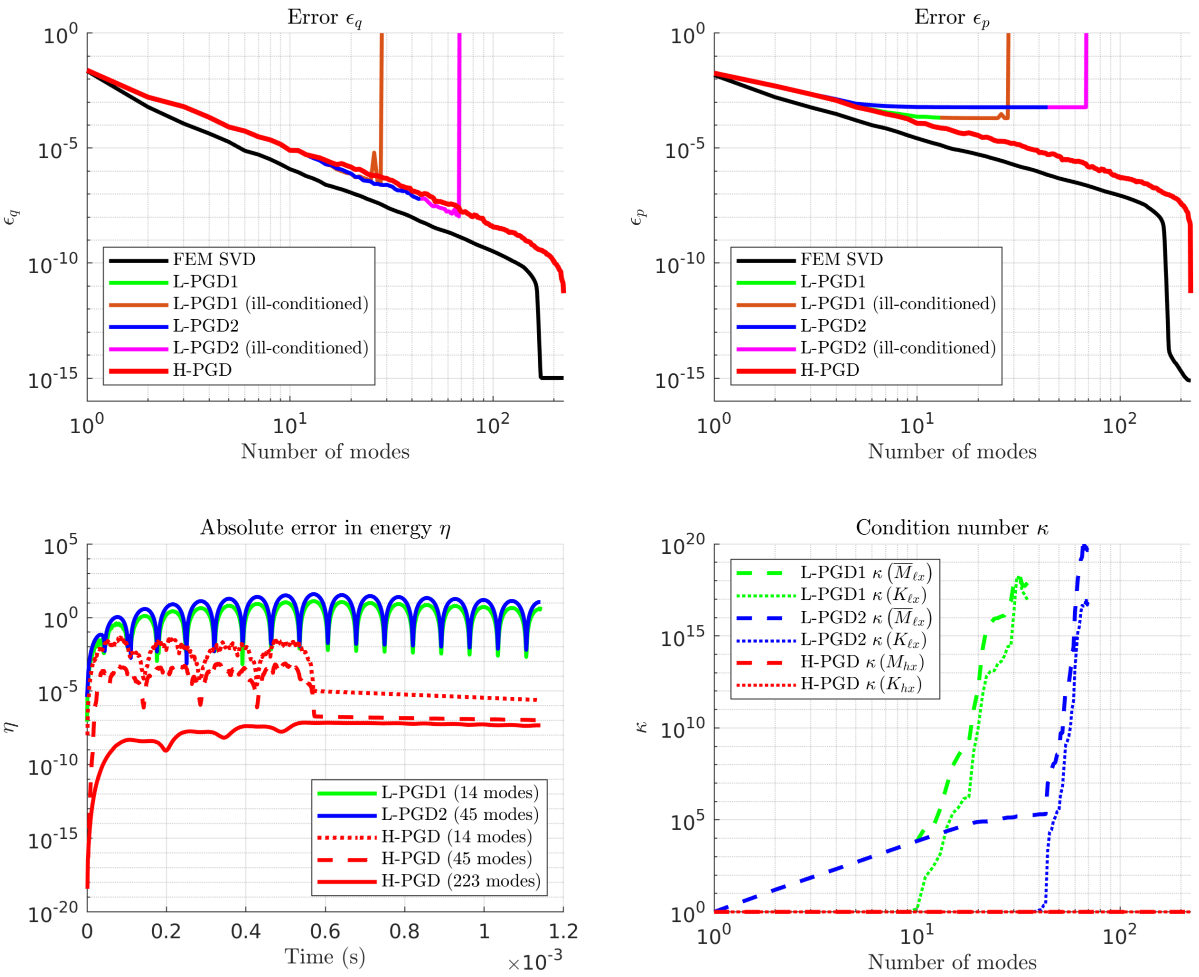


Figure 2.13 Case 5. (Top left) Error between the reference displacement field and the SVD or PGD displacement field. (Top right) Error between the reference conjugate momenta field and the SVD or PGD conjugate momenta field. (Bottom left) Error between the energy of the reference system and the energy of the reduced system over time. (Bottom right) Condition numbers of the matrices introduced in Section 2.3.3.

The last case considers the exact same scenario presented in Section 2.4.4, but for the presence of an extra linear damping term in the wave equation, i.e.:

$$\rho A \frac{\partial^2 u}{\partial t^2} + \zeta \frac{\partial u}{\partial t} - EA \frac{\partial^2 u}{\partial x^2} = 0, \qquad \forall (x,t) \in \Omega \times \mathcal{I},$$

where $\zeta = 15 \times 10^3$ Pl (1 Poiseuille = 1 kg/m/s) is the so-called damping coefficient.

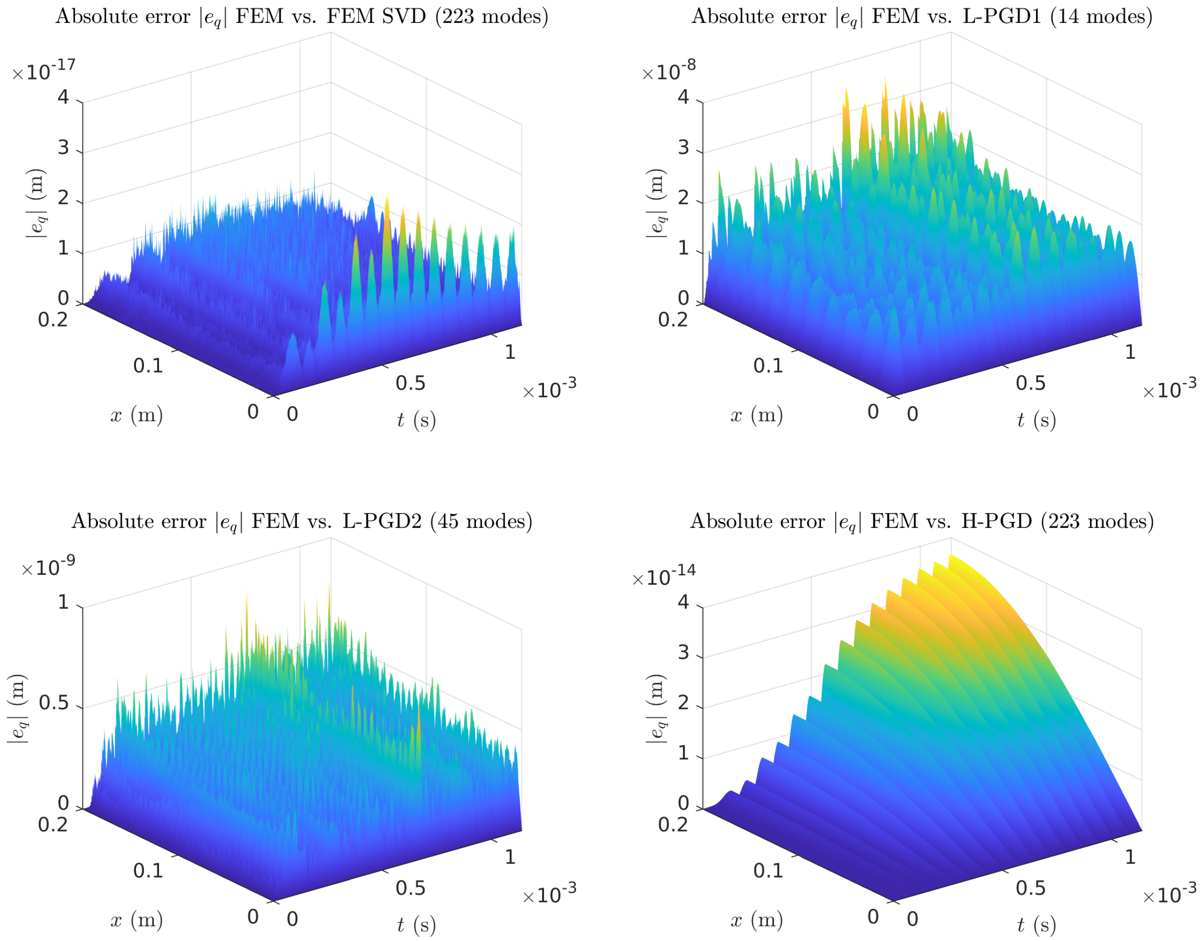


Figure 2.14 Case 5. Absolute errors in space and time between the reference displacement field and the SVD or PGD displacement field.

The results are shown in Figures 2.13, 2.14, and 2.15. These are qualitatively very similar to those presented in Section 2.4.4, except that the total energy of the bar decreases after $t \geq T/2$, as expected. The main objective of this example is to illustrate that the H-PGD framework is also suitable for the study of linear elasticity problems accounting for energy dissipation. First, the H-PGD model reduction provides better stability and energy conservation of the original system than the L-PGD approaches. Moreover, reduced-order modeling methods for problems with damping based on modal decomposition [15] lead to an eigenvalue problem that requires a more elaborated treatment [68, 91] than the eigenvalue problem obtained without damping. The Rayleigh hypothesis is often used to circumvent the issue introduced by the damping matrix [68]. However, the hypothesis does not have an unequivocal physical meaning [69, 70] and may produce underdamped or overdamped behaviors in certain frequency ranges [68] (although this may be convenient in some cases). In contrast to the modal decomposition, it is not necessary in the PGD framework to resort

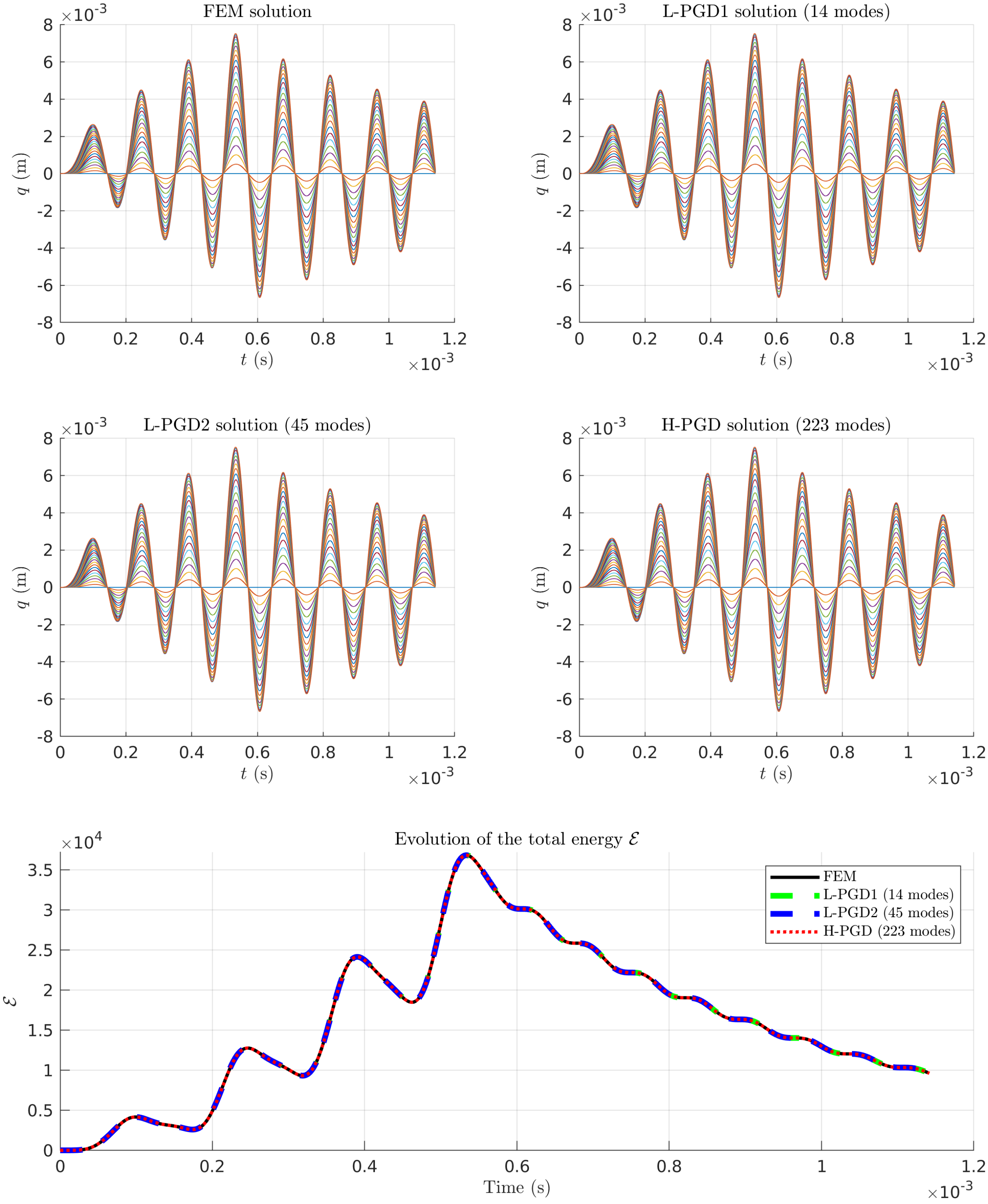


Figure 2.15 Case 5. (Top four plots) Evolution of the displacement field over time for the different reduction methods; the displacements are shown at 23 nodes uniformly distributed along the bar. (Bottom) Evolution of the total energy of the reference and reduced systems versus time.

to a special treatment in order to account for the damping term in the wave equation.

### 2.4.8 Further discussion

Non-symmetry and ill-conditioning of the matrices are issues that are also mentioned in [55, Pages 4 and 15]. The authors briefly discuss the eventual ill-conditioning of the operator $\mathbf{A}$ (non-symmetric), which represents the discretization of the space-time bilinear form of the problem. This operator is constant and its conditioning results from the fineness of the space-time discretization and the mechanical properties of the problem ($E$, $\rho$, etc.). In our case, the matrices $K_{\ell x}$, $M_{\ell x}$, $\bar{M}_{\ell x}$, $K_{hx}$ and $M_{hx}$ (Gram matrices) not only depend on the discretization and mechanical properties but are also non-constant. Indeed, their size grows with the number of enrichment $m$. Their ill-conditioning results mainly from the fact that the basis vectors may become linearly dependent. In conclusion, the nature of the matrices we studied is not the same as that of $\mathbf{A}$ and the reasons of their bad conditioning are different.

From a numerical point of view, we would like to emphasize that the only numerical difference between the Lagrangian update (2.42) and the Hamiltonian one (2.43) is the scaling factor $\rho A$. One may wonder whether this factor has an influence on the convergence of the H-PGD solver and the conditioning of the system. Multiple tests were run in the case where all parameters of the problem were set to unity: $E = \rho = A = \ell = 1$. In this case, the scaling factor is equal to unity and the systems (2.42) and (2.43) are numerically identical. However, H-PGD remains numerically stable and still provides more accurate solutions than L-PGD. Therefore, it is not just the Hamiltonian formulation as a standalone formalism that enables improvements over the Lagrangian approach, but also the new possibilities that it offers in terms of algorithmic design.

Moreover, it is worth noting that the numerical values of the Young's modulus $E$ and the density $\rho$ play a major role in the non-symmetry. The non-symmetry is due to the temporal derivation. In other words, non-symmetry dominates if inertial effects are preponderant, i.e. when $\rho > E$ or, similarly, when the velocity $c = \sqrt{E/\rho}$ is small. One can find the same reasoning with the Heat Equation and the thermal diffusivity in [85] (Page 68). Yet, in structural dynamics, $E \gg \rho$ in general. For instance, in our test cases, the material properties are chosen as those of steel with $E = 220$ GPa, $\rho = 7000$ kg/m$^3$, similarly to [55]. These numerical values are actually advantageous because they are not favorable to non-symmetry. Thus, the case of unitary parameters also enabled us to test our algorithm in examples where matrices become asymmetric and validate its robustness.

Energy conservation is related to the time integrators. Nevertheless, the computed PGD modes also play a significant role. This is best illustrated in the case of the problem with a

Neumann boundary condition (see Figures 2.4 and 2.13). In particular, we observe on the top right plot that after the 5th enrichment, L-PGD fails at recovering the modes with respect to the momentum field (i.e., error $\epsilon_p$). This results in a bad energy conservation. We conclude from these results that energy conservation is also sensitive to the PGD formulation and that H-PGD performs better that L-PGD by several orders of magnitude (as shown on the bottom left plot of the figures).

## 2.5 Conclusion

Galerkin-based PGD formulations based on the Hamilton's weak principle have been developed to derive reduced-order models of second-order hyperbolic systems. One of the objectives was in particular to compute a reduced model that preserves the energy of the system by means of stable, energy conservative integration schemes. We have considered in this work two approaches, namely the L-PDG and the H-PGD. The former is based on the Lagrangian formalism while the latter follows from the Hamiltonian formalism. The H-PGD approach describes the system dynamics in terms of the generalized coordinates and the generalized momenta. The two fields can then be represented as two distinct expansions whose modes are solutions of coupled problems. The procedure brings some flexibility, in particular, it enables one to design orthogonalization and updating processes that ensure computational stability. Moreover, we have designed an adaptive fixed-point algorithm for the H-PGD approach that controls the convergence of the two fields separately. The combination of these procedures does improve convergence and eliminates redundancy in the enrichment modes of the H-PDG approach. As a result, for the test cases considered here, the H-PGD formulation showed a much better behavior in terms of stability and energy preservation than the L-PGD formulation.

Nevertheless, the 1D problems that were tackled in this chapter were very small and the use of a reduced-order method cannot be justified. Actually, the PGD algorithm that was developed herein was computationally inefficient and exhibited a poor scalability. Furthermore, the question of the preservation of the symplectic structure in the context of Hamiltonian systems was not tackled. These aspects will be the object of the next chapter.

# CHAPTER 3 ON AN EFFICIENT, SYMPLECTICITY-PRESERVING SPACE-TIME PROPER GENERALIZED DECOMPOSITION

The relevance of a reduced-order modeling technique stems from its ability to exceed the computational efficiency of a conventional Finite Element model, while incurring a relatively low error with respect to the FE solution of the full model. So far, if space-time PGD solvers have demonstrated a satisfying level of accuracy with a rather low number of modes, their computational efficiency is far from being competitive [50]. In this chapter, we develop a novel space-time PGD solver with a focus on computational efficiency. The integration of the PGD strategy within the Hamiltonian formalism is revisited and we comment on the preservation of the symplectic structure on the time parameter by the reduced model. The Aitken transformation [92] has subsequently been introduced to accelerate the convergence of the fixed-point algorithm. We will show that it significantly reduces the number of required iterations for convergence. Additionally, a new orthogonal projection, more robust than the one formerly implemented, is performed on the spatial modes to enforce their linear independence and ensure the stability of the algorithm. Yet, the computational cost of such solvers mainly depends on the problem with respect to the spatial variable, which needs to be assembled and factorized at each fixed-point iteration. An original approach has been developed to avoid having to repeatedly factorize matrices. It consists in pre-processing the eigen-pair approximations of the operators, namely the Ritz pairs [16], that provide a subspace in which the problem in space remains diagonal throughout the fixed-point iterations. In the manner of Modal Decomposition, all computations are then carried out in the subspace spanned by the Ritz vectors [93], hence drastically decreasing the computational burden while capturing using only a small number of modes most of the information from the full model. Numerical examples dealing with the dynamical behavior of a 3D structure will be presented in order to demonstrate the efficiency of the proposed approach.

The chapter is organized as follows: in Section 3.1, we describe the model problem and its spatial Finite Element approximation. In Section 3.2, we present the Hamiltonian formalism and its symplectic structure. The PGD approaches are described in Section 3.3 along with the Aitken acceleration and the orthogonal projectors applied to the fixed-point algorithm, as well as the projection of the PGD approximation onto the subspace spanned by the Ritz vectors. The numerical experiments are presented in Section 3.4 to illustrate the performance of the proposed approach. We finally provide some concluding remarks in Section 3.5.

## 3.1 Model problem

### 3.1.1 Strong formulation

The model problem we shall consider is that of elastodynamics in three dimensions under the assumption of infinitesimal deformation. Let $\Omega$ be an open bounded subset of $\mathbb{R}^3$, with Lipschitz boundary $\partial\Omega$, and let $\mathcal{I} = (0, T)$ denote the time interval. The boundary $\partial\Omega$ is supposed to be decomposed into two portions, $\partial\Omega_D$ and $\partial\Omega_N$, such that $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. The displacement field $u : \bar{\Omega} \times \bar{\mathcal{I}} \to \mathbb{R}^3$ satisfies the following partial differential equation:

$$\rho\frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma(u) = f, \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{3.1}$$

where, in the case of infinitesimal deformation, the stress tensor $\sigma(u)$ and strain tensor $\varepsilon(u)$ are given by:

$$\sigma(u) = \mathbb{E} : \varepsilon(u), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{3.2}$$

$$\varepsilon(u) = \frac{1}{2}\left(\nabla u + \left(\nabla u\right)^T\right), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{3.3}$$

and is subjected to the initial conditions:

$$u(x,0) = u_0(x), \qquad \forall x \in \Omega, \tag{3.4}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \qquad \forall x \in \Omega, \tag{3.5}$$

as well as to the boundary conditions:

$$u(x,t) = 0, \qquad \forall (x,t) \in \partial\Omega_D \times \mathcal{I}, \tag{3.6}$$

$$\sigma(u) \cdot n = g_N(x,t), \qquad \forall (x,t) \in \partial\Omega_N \times \mathcal{I}. \tag{3.7}$$

The functions $f : \Omega \times \mathcal{I} \to \mathbb{R}^3$, $u_0 : \Omega \to \mathbb{R}^3$, $v_0 : \Omega \to \mathbb{R}^3$, and $g_N : \partial\Omega_N \times \mathcal{I} \to \mathbb{R}^3$ are supposed to be sufficiently regular to yield a well-posed problem. The medium occupied by $\bar{\Omega}$ is assumed to be isotropic, with density $\rho$ and Lamé coefficients $\lambda$, $\mu$ (the material parameters could possibly vary in space). The constitutive equation (3.2), written above in terms of the tensor of elasticity $\mathbb{E}$, thus reduces to:

$$\sigma(u) = 2\mu\varepsilon(u) + \lambda\mathrm{tr}\left(\varepsilon(u)\right) I_3,$$

where $I_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix. In the following, we will denote the first and second time derivatives by $\dot{u} = \partial u/\partial t$ and $\ddot{u} = \partial^2 u/\partial t^2$.

### 3.1.2 Semi-weak formulation

We consider here the semi-weak formulation with respect to the spatial variable in order to construct the discrete problem in space using the Finite Element method. Multiplying (3.1) by an arbitrary smooth function $u^* = u^*(x)$ and integrating over the whole domain $\Omega$, one obtains:

$$\int_\Omega \rho\ddot{u}\cdot u^* - (\nabla\cdot\sigma(u))\cdot u^*\ dx = \int_\Omega f\cdot u^*\ dx, \qquad \forall t \in \mathcal{I}. \tag{3.8}$$

By virtue of $-(\nabla\cdot\sigma(u))\cdot u^* = \sigma(u) : \nabla u^* - \nabla\cdot(\sigma(u)\cdot u^*)$, Eq. (3.8) can be recast as:

$$\int_\Omega \rho\ddot{u}\cdot u^* + \sigma(u) : \nabla u^*\ dx = \int_\Omega \nabla\cdot(\sigma(u)\cdot u^*)\ dx + \int_\Omega f\cdot u^*\ dx, \qquad \forall t\in\mathcal{I}.$$

Since $\sigma(u)$ is a symmetric tensor:

$$\sigma(u) : \nabla u^* = \sigma(u) : \varepsilon(u^*),$$

and substituting the constitutive equation for $\sigma(u)$, one gets:

$$\sigma(u) : \varepsilon(u^*) = (\mathbb{E} : \varepsilon(u)) : \varepsilon(u^*) = \varepsilon(u) : \mathbb{E} : \varepsilon(u^*).$$

Applying the divergence theorem and the boundary conditions, and choosing the test function such that $u^* = 0$ on $\partial\Omega_D$, the semi-discrete formulation of the problem then reads: Find $u = u(\cdot, t) \in V$, for all $t \in \bar{\mathcal{I}}$, such that:

$$\int_\Omega \rho\ddot{u}\cdot u^* + \varepsilon(u) : \mathbb{E} : \varepsilon(u^*)\ dx = \int_\Omega f\cdot u^*\ dx + \int_{\partial\Omega_N} g_N\cdot u^*\ dx, \qquad \forall u^*\in V, \quad \forall t\in\mathcal{I}, \tag{3.9}$$

and:

$$u(x,0) = u_0(x), \qquad \forall x\in\Omega, \tag{3.10}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \qquad \forall x\in\Omega, \tag{3.11}$$

where $V$ is the vector space of vector-valued functions defined on $\Omega$:

$$V = \left\{ v \in \left[H^1(\Omega)\right]^3 :\ v = 0 \text{ on } \partial\Omega_D \right\}.$$

### 3.1.3 Spatial discretization

We partition the domain into $N_e$ elements, denoted by $K_e$, such that $\overline{\Omega} = \cup_{e=1}^{N_e} K_e$ and $\mathrm{Int}(K_i) \cap \mathrm{Int}(K_j) = \varnothing$, $\forall i, j = 1, \ldots, N_e$, $i \neq j$. We then associate with the mesh the finite-dimensional Finite Element space $W^h$, $\dim W^h = s$, of scalar-valued continuous and piecewise polynomial functions defined on $\Omega$, that is:

$$W^h = \left\{ v_h : \Omega \to \mathbb{R} : \ v_h|_{K_e} \in \mathbb{P}_k(K_e), \ e = 1, \ldots, N_e \right\},$$

where $\mathbb{P}_k(K_e)$ denotes the space of polynomial functions of degree $k$ on $K_e$. Let $\phi_i$, with $i = 1, \ldots, s$, denote the basis functions of $W^h$, i.e. $W^h = \mathrm{span}\{\phi_i\}$. We then introduce the finite element subspace $V^h$ of $V$ such that:

$$V^h = \left[ W^h \right]^3 \cap V,$$

and search for finite element solutions $u_h = u_h(\cdot, t) \in V^h$, $\forall t \in \bar{\mathcal{I}}$, in the form:

$$u_h(x,t) = \sum_{j=1}^{s} q_j(t) \phi_j(x),$$

where the vectors of degrees of freedom, $q_j \in \mathbb{R}^3$, depend on time. We introduce the set of $n = 3s$ vector-valued basis functions as:

$$\chi_{3i-2}(x) = \begin{bmatrix} \phi_i(x) \\ 0 \\ 0 \end{bmatrix}, \quad \chi_{3i-1}(x) = \begin{bmatrix} 0 \\ \phi_i(x) \\ 0 \end{bmatrix}, \quad \chi_{3i}(x) = \begin{bmatrix} 0 \\ 0 \\ \phi_i(x) \end{bmatrix}, \qquad i = 1, \ldots, s.$$

Using the Galerkin method, the Finite Element problem thus reads:

Find $u_h(\cdot, t) \in V^h$, such that

$$\begin{aligned} &\int_\Omega \rho \chi_i(x) \cdot \ddot{u}_h(x,t) + \varepsilon(\chi_i)(x) : \mathbb{E} : \varepsilon(u_h)(x,t)\, dx \\ &\qquad = \int_\Omega \chi_i(x) \cdot f(x,t)\, dx + \int_{\partial\Omega_N} g_N(x,t) \cdot \chi_i(x)\, ds, \quad \forall i = 1, \ldots, n, \quad \forall t \in \mathcal{I}, \end{aligned}$$

satisfying the initial conditions

$$\begin{aligned} u_h(x,0) &= u_{0,h}(x), \quad \forall x \in \Omega, \\ \dot{u}_h(x,0) &= v_{0,h}(x), \quad \forall x \in \Omega, \end{aligned}$$

where $u_{0,h}$ and $v_{0,h}$ are interpolants or projections of $u_0$ and $v_0$ in the space $V^h$. The above problem can be conveniently recast in compact form as:

$$M\ddot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{3.12}$$

$$\boldsymbol{q}(0) = \boldsymbol{u}_0, \tag{3.13}$$

$$\dot{\boldsymbol{q}}(0) = \boldsymbol{v}_0, \tag{3.14}$$

where $M$ and $K$ are the global mass and stiffness matrices, respectively, both being symmetric and positive definite:

$$M_{ij} = \int_\Omega \rho\chi_i \cdot \chi_j \; dx, \qquad K_{ij} = \int_\Omega \varepsilon(\chi_i) : \mathbb{E} : \varepsilon(\chi_j) \; dx, \qquad \forall i, j = 1, \ldots, n,$$

$\boldsymbol{f}(t)$ is the load vector at time $t$ whose components are given by:

$$f_i(t) = \int_\Omega \chi_i(x) \cdot f(x,t)\, dx + \int_{\partial\Omega_N} \chi_i(x) \cdot g_N(x,t) \; ds, \qquad \forall i = 1, \ldots, n,$$

$\boldsymbol{q}(t)$ is the global vector of degrees of freedom:

$$\boldsymbol{q}(t) = \begin{bmatrix} q_1(t) & \ldots & q_s(t) \end{bmatrix}^T,$$

and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial vectors:

$$\boldsymbol{u}_0 = \begin{bmatrix} u_{0,1} & \ldots & u_{0,s} \end{bmatrix}^T,$$
$$\boldsymbol{v}_0 = \begin{bmatrix} v_{0,1} & \ldots & v_{0,s} \end{bmatrix}^T.$$

Note that $u_{0,i} \in \mathbb{R}^3$ and $v_{0,i} \in \mathbb{R}^3$, $i = 1, \ldots, s$, are vectors whose components are the initial displacements and velocities in the three spatial directions.

## 3.2 The Hamiltonian formalism

### 3.2.1 Hamilton's Weak Principle

The Hamiltonian formalism consists in modeling the motion of the system along a trajectory in the phase space by introducing the generalized coordinates $\boldsymbol{q}$ and their generalized (or conjugate) momenta $\boldsymbol{p}$ as independent variables. For the problem at hand, the Hamiltonian functional $h$ reads:

$$h(\boldsymbol{q}, \boldsymbol{p}, t) = \frac{1}{2}\boldsymbol{q}^T K \boldsymbol{q} + \frac{1}{2}\boldsymbol{p}^T M^{-1} \boldsymbol{p} - \boldsymbol{q}^T \boldsymbol{f}. \tag{3.15}$$

Given the Hamiltonian functional $h$ of the system, the action functional, denoted by $\mathcal{S}[\boldsymbol{q},\boldsymbol{p}]$, is defined as:

$$\mathcal{S}[\boldsymbol{q},\boldsymbol{p}] = \int_{\mathcal{I}} \dot{\boldsymbol{q}}^T\boldsymbol{p} - h(\boldsymbol{q},\boldsymbol{p},t)\ dt$$

The Hamilton's Weak Principle then states that the trajectory $(\boldsymbol{q},\boldsymbol{p})$ of the system in the phase space should satisfy:

$$\mathcal{S}'[\boldsymbol{q},\boldsymbol{p}](\boldsymbol{q}^*,\boldsymbol{p}^*) = \left[\boldsymbol{q}^{*T}\boldsymbol{p}\right]_0^T,$$

where $\mathcal{S}'[\boldsymbol{q},\boldsymbol{p}](\boldsymbol{q}^*,\boldsymbol{p}^*)$ denotes the Gâteaux derivative of $\mathcal{S}[\boldsymbol{q},\boldsymbol{p}]$ with respect to a variation $(\boldsymbol{q}^*,\boldsymbol{p}^*) \in \mathcal{Z}\times\mathcal{Z}$ such that:

$$\mathcal{Z} = \left\{\boldsymbol{v} \in [C^1(\bar{\mathcal{I}})]^n;\ \boldsymbol{v}(0) = 0\right\}.$$

After Gâteaux derivation and integration by parts with respect to time, we get:

$$\int_{\mathcal{I}} \boldsymbol{p}^{*T}\dot{\boldsymbol{q}} - \boldsymbol{q}^{*T}\dot{\boldsymbol{p}} - \boldsymbol{q}^{*T}K\boldsymbol{q} - \boldsymbol{p}^{*T}M^{-1}\boldsymbol{p} + \boldsymbol{q}^{*T}\boldsymbol{f}\ dt = 0, \qquad \forall(\boldsymbol{q}^*,\boldsymbol{p}^*) \in \mathcal{Z}\times\mathcal{Z},$$

that is,

$$\int_{\mathcal{I}} \boldsymbol{q}^{*T}\dot{\boldsymbol{p}} - \boldsymbol{p}^{*T}\dot{\boldsymbol{q}} + \boldsymbol{q}^{*T}K\boldsymbol{q} + \boldsymbol{p}^{*T}M^{-1}\boldsymbol{p}\ dt = \int_{\mathcal{I}} \boldsymbol{q}^{*T}\boldsymbol{f}\ dt, \qquad \forall(\boldsymbol{q}^*,\boldsymbol{p}^*) \in \mathcal{Z}\times\mathcal{Z},$$

or, equivalently,

$$\begin{aligned} &\int_{\mathcal{I}} \boldsymbol{q}^{*T}\left(\dot{\boldsymbol{p}} + K\boldsymbol{q}\right)\ dt = \int_{\mathcal{I}} \boldsymbol{q}^{*T}\boldsymbol{f}\ dt, && \forall\boldsymbol{q}^* \in \mathcal{Z},\\ &\int_{\mathcal{I}} \boldsymbol{p}^{*T}\left(\dot{\boldsymbol{q}} - M^{-1}\boldsymbol{p}\right)\ dt = 0, && \forall\boldsymbol{p}^* \in \mathcal{Z}. \end{aligned} \tag{3.16}$$

The last weak formulation of (3.16) leads the so-called Hamilton's equations:

$$\begin{aligned} \dot{\boldsymbol{p}} + K\boldsymbol{q} &= \boldsymbol{f},\\ \dot{\boldsymbol{q}} - M^{-1}\boldsymbol{p} &= \boldsymbol{0}. \end{aligned} \tag{3.17}$$

This formulation is consistent with (3.12) in the sense that if we differentiate with respect to time the second equation and substitute $\dot{\boldsymbol{p}}$ for the expression in the first equation, we do exactly recover (3.12).

### 3.2.2 Symplectic structure

Let us introduce $\boldsymbol{z} \in \mathcal{Z}^2$ that vertically concatenates $\boldsymbol{q}$ and $\boldsymbol{p}$ such that:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}.$$

The gradient of the Hamiltonian (3.15) then reads:

$$\nabla_z h = \begin{bmatrix} \nabla_q h \\ \nabla_p h \end{bmatrix} = \begin{bmatrix} K\boldsymbol{q} - \boldsymbol{f} \\ M^{-1}\boldsymbol{p} \end{bmatrix}.$$

In the symplectic framework, the dynamics of the structure is modeled by the trajectory in the symplectic vector space $(\mathbb{R}^{2n}, \omega)$ of dimension $2n$ for linear systems, where $\omega$ is the so-called symplectic form defined as:

$$\forall \boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix} \in \mathbb{R}^{2n}, \quad \forall \boldsymbol{z}' = \begin{bmatrix} \boldsymbol{q}' \\ \boldsymbol{p}' \end{bmatrix} \in \mathbb{R}^{2n}, \quad \omega(\boldsymbol{z}, \boldsymbol{z}') = \boldsymbol{q}^T \boldsymbol{p}' - \boldsymbol{q}'^T \boldsymbol{p} = \boldsymbol{z}^T J_{2n} \boldsymbol{z}',$$

with $J_{2n}$ the skew-symmetric operator such that:

$$J_{2n} = \begin{bmatrix} 0 & I_n \\ -I_n & 0 \end{bmatrix},$$

and $J_{2n}^2 = -I_{2n}$. It is then possible to recast (3.17) as:

$$\dot{\boldsymbol{z}} = \nabla^\omega h,$$

where $\nabla^\omega = J_{2n} \nabla_z$ is defined as the symplectic gradient. The Hamiltonian can be written as a sum of a quadratic form on $\mathbb{R}^{2n}$ and the external energy term:

$$h(\boldsymbol{z}, t) = \frac{1}{2} \boldsymbol{z}^T H \boldsymbol{z} - \boldsymbol{z}^T \boldsymbol{f}_z,$$

with $H$ the Hessian operator of $h$ and $\boldsymbol{f}_z$ such that:

$$H = \begin{bmatrix} K & 0 \\ 0 & M^{-1} \end{bmatrix}, \qquad \boldsymbol{f}_z = \begin{bmatrix} \boldsymbol{f} \\ 0 \end{bmatrix}.$$

It follows that one can rewrite the weak formulation (3.16) as:

$$\int_{\mathcal{I}} \boldsymbol{z}^{*T} \left[ J_{2n} \dot{\boldsymbol{z}} + H \boldsymbol{z} \right] \, dt = \int_{\mathcal{I}} \boldsymbol{z}^{*T} \boldsymbol{f}_z \, dt, \qquad \forall \boldsymbol{z}^* \in \mathcal{Z}^2. \tag{3.18}$$

We now introduce the notion of symplectic mapping. A symplectic mapping is a linear transformation that preserves the symplectic form $\omega$, i.e.:

$$A \in \mathbb{R}^{2n \times 2n} \text{ is symplectic if } \quad \omega(A\boldsymbol{z}, A\boldsymbol{z}') = \omega(\boldsymbol{z}, \boldsymbol{z}'), \quad \forall (\boldsymbol{z}, \boldsymbol{z}') \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}.$$

As a consequence, such a mapping $A$ verifies:

$$A^T J_{2n} A = J_{2n}.$$

The notion can actually be generalized to rectangular matrices with the symplectic Stiefel manifold, denoted $S_p(2r, 2n)$, such that:

$$S_p(2r, 2n) = \left\{ A \in \mathbb{R}^{2n \times 2r} : \; A^T J_{2n} A = J_{2r} \right\} \tag{3.19}$$

Let $(\mathbb{R}^{2r}, \gamma)$ be a symplectic vector space, $A \in S_p(2r, 2n)$ a symplectic mapping, and $\boldsymbol{y} \in \mathbb{R}^{2r}$ such that $\boldsymbol{z} = A\boldsymbol{y}$. One can define a Hamiltonian for $\boldsymbol{y}$:

$$g(\boldsymbol{y}) = \frac{1}{2} \boldsymbol{y}^T G \boldsymbol{y} - \boldsymbol{y}^T \boldsymbol{f}_y,$$

with $G$ its Hessian operator and $\boldsymbol{f}_y$ the projection of the external loads on the symplectic subspace (in the case $r \leqslant n$), such that:

$$G = A^T H A, \quad \text{and} \quad \boldsymbol{f}_y = A^T \boldsymbol{f}_z.$$

The preservation of the symplectic structure implies that $\boldsymbol{y}$ is governed by Hamilton's canonical equations, expressed hereinafter in terms of $\gamma$ (symplectic form on $\mathbb{R}^{2r}$) and $g$ such that:

$$\dot{\boldsymbol{y}} = \nabla^{\gamma} g,$$

with $\nabla^{\gamma} = J_{2r} \nabla_y$ and Hamilton's weak principle (3.16) then reads:

$$\int_{\mathcal{I}} \boldsymbol{y}^{*T} \left[ J_{2r} \dot{\boldsymbol{y}} + G \boldsymbol{y} \right] \, dt = \int_{\mathcal{I}} \boldsymbol{y}^{*T} \boldsymbol{f}_y \, dt, \qquad \forall \boldsymbol{y}^* \in \mathbb{R}^{2r}.$$

### 3.2.3 Discretization in time of the Hamiltonian problem

The time domain $\mathcal{I}$ is divided into $n_t$ subintervals $\mathcal{I}^i = [t^{i-1}, t^i]$, $i = 1, \dots, n_t$, of size $h_t = t^i - t^{i-1}$. The Crank-Nicolson method is then applied to (3.17) as detailed in the previous work [94]. The solutions given by the FEM in space, integrated with Crank-Nicolson in time, will be used as reference solutions when assessing the results of the PGD solvers.

Although not the primary focus of this article, we acknowledge the relevance of symplectic integrators in the case of Hamiltonian mechanics. These integrators are particularly robust to compute long-time evolution of Hamiltonian systems [37–39]. In addition, the preservation of the symplectic structure by the reduced model is the subject of numerous studies [40–42]. We will also discuss this property on the time parameter with respect to our PGD solver in Section 3.3.3.

## 3.3 PGD reduced-order modeling

The proper-generalized decomposition method applied within the Hamiltonian framework aims at approximating both the generalized coordinates $\boldsymbol{q}$ and their generalized momenta $\boldsymbol{p}$ in separated form. We are thus searching for a space-time separated representation of $\boldsymbol{z}$ as:

$$\boldsymbol{z}(t) \simeq \boldsymbol{z}_m(t) = \sum_{i=1}^{m} \Phi_i \boldsymbol{\psi}_i(t) = \sum_{i=1}^{m} \Psi_i(t) \boldsymbol{\varphi}_i,$$

with:

$$\Phi_i = \begin{bmatrix} \boldsymbol{\varphi}_i^q & 0 \\ 0 & \boldsymbol{\varphi}_i^p \end{bmatrix}, \qquad \boldsymbol{\psi}_i = \begin{bmatrix} \psi_i^q \\ \psi_i^p \end{bmatrix},$$
$$\Psi_i = \begin{bmatrix} \psi_i^q I_n & 0 \\ 0 & \psi_i^p I_n \end{bmatrix}, \qquad \boldsymbol{\varphi}_i = \begin{bmatrix} \boldsymbol{\varphi}_i^q \\ \boldsymbol{\varphi}_i^p \end{bmatrix},$$

where $\Phi_i$ is a $(2n \times 2)$ matrix and $\boldsymbol{\psi}_i$ a $(2 \times 1)$ vector while $\Psi_i$ is a $(2n \times 2n)$ matrix and $\boldsymbol{\varphi}_i$ a $(2n \times 1)$ vector. The two notations are mathematically equivalent and convenient whether the weak formulation is solved for $\boldsymbol{\varphi}$ (spatial problem) or $\boldsymbol{\psi}$ (temporal problem). The vector-valued functions $(\boldsymbol{\varphi}_i^q)_{1 \leqslant i \leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1 \leqslant i \leqslant m}$ provide the spatial bases for the generalized coordinates and conjugate momenta, respectively:

$$\boldsymbol{q}(t) \approx \boldsymbol{q}_m(t) = \sum_{i=1}^{m} \boldsymbol{\varphi}_i^q \psi_i^q(t),$$
$$\boldsymbol{p}(t) \approx \boldsymbol{p}_m(t) = \sum_{i=1}^{m} \boldsymbol{\varphi}_i^p \psi_i^p(t).$$

For the sake of clarity in the presentation, we shall drop from now on the subscript $i$ and write the decomposition of rank $m$ of $\boldsymbol{z}$ as:

$$\boldsymbol{z}_m(t) = \boldsymbol{z}_{m-1}(t) + \Phi\boldsymbol{\psi}(t), \qquad \text{or} \qquad \boldsymbol{z}_m(t) = \boldsymbol{z}_{m-1}(t) + \Psi(t)\boldsymbol{\varphi}.$$

The approach considered here is the so-called greedy rank-one update algorithm, where the separated representation is computed progressively by adding one pair of modes $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$ at each enrichment. The goal in this section is to construct the separated spatial and temporal problems that satisfy the enrichment modes $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$, the new unknowns of the problem, assuming that the previous iterate $\boldsymbol{z}_{m-1}$ has already been calculated.

### 3.3.1 Fixed-point strategy

Computing a separated representation of $\boldsymbol{q}$ and $\boldsymbol{p}$ demands an adequate solution strategy of the weak formulation (3.18). Substituting the trial solution $\boldsymbol{z}_m$ for $\boldsymbol{z}$ in (3.18) leads to a non-linear formulation for the modes $\boldsymbol{\varphi}$ and $\boldsymbol{\psi}$. Several iterative schemes could be used to solve such a problem. The fixed point algorithm is considered here, which proceeds as follows:

1. Solve (3.18) for $\boldsymbol{\varphi}$ with $\boldsymbol{\psi}$ known. This step is referred to as the spatial problem and is written in a generic format as:

$$A(\boldsymbol{\psi})\boldsymbol{\varphi} = \boldsymbol{b}(\boldsymbol{\psi}, \boldsymbol{z}_{m-1}), \tag{3.20}$$

where the matrix $A(\boldsymbol{\psi})$ and vector $\boldsymbol{b}(\boldsymbol{\psi}, \boldsymbol{z}_{m-1})$ will be specified in Section Problem in space. More precisely, in order to enhance robustness, we propose to force the new spatial mode to preserve the linear independence of the spatial bases $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m}$, which can formally be written as:

$$\boldsymbol{\varphi} = P_m A(\boldsymbol{\psi})^{-1}\boldsymbol{b}(\boldsymbol{\psi}, \boldsymbol{z}_{m-1})$$

where $P_m$ is a projector that is orthogonal to the subspace spanned by previous mode (for a well chosen inner product).

2. Solve (3.18) for $\boldsymbol{\psi}$ with $\boldsymbol{\varphi}$ known. The temporal problem corresponds to the system of first-order differential equations:

$$\dot{\boldsymbol{\psi}} = f_{\mathcal{T}}(\boldsymbol{\psi}, \boldsymbol{\varphi}, \boldsymbol{z}_{m-1}), \tag{3.21}$$

where the vector-valued function $f_{\mathcal{T}}$ will be explicitly provided in Section Problem in time.

Steps 1 and 2 are repeated until a convergence criterion is fulfilled. It is noteworthy that (3.20) is a linear system of size $2n$ associated with the space discretization, similar to that of a steady-state FEM problem. Eq. (3.21) is a system of two first order scalar ordinary differential equations in time, solved for $\psi_q$ and $\psi_p$. Both problems are described in the next sections.

### Problem in space

We assume that $\boldsymbol{\psi}$ is known and search for the new spatial mode $\boldsymbol{\varphi}$. We substitute $\boldsymbol{z}_{m-1} + \Psi\boldsymbol{\varphi}$ for $\boldsymbol{z}$ in (3.18) and choose test functions in the form $\boldsymbol{z}^* = \Psi\boldsymbol{\varphi}^*$. Equation (3.18) reduces to:

$$\int_{\mathcal{I}} \boldsymbol{\varphi}^{*T} \Psi^T \left( J_{2n} \dot{\Psi} \boldsymbol{\varphi} + H \Psi \boldsymbol{\varphi} \right) \, dt = \int_{\mathcal{I}} \boldsymbol{\varphi}^{*T} \Psi^T \left( \boldsymbol{f} - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1} \right) \, dt, \qquad \forall \boldsymbol{\varphi}^* \in \mathbb{R}^{2n},$$

which, since $\boldsymbol{\varphi}^*$ and $\boldsymbol{\varphi}$ are independent of time, can be rewritten as:

$$\boldsymbol{\varphi}^{*T} \left[ \int_{\mathcal{I}} \Psi^T J_{2n} \dot{\Psi} + \Psi^T H \Psi \, dt \right] \boldsymbol{\varphi} = \boldsymbol{\varphi}^{*T} \left[ \int_{\mathcal{I}} \Psi^T \left( \boldsymbol{f} - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1} \right) \, dt \right], \quad \forall \boldsymbol{\varphi}^* \in \mathbb{R}^{2n}.$$

This leads to the following linear system:

$$A_{\mathcal{S}} \boldsymbol{\varphi} = \boldsymbol{b}_{\mathcal{S}}, \tag{3.22}$$

with:

$$A_{\mathcal{S}} = \left[ \int_{\mathcal{I}} \Psi^T J_{2n} \dot{\Psi} + \Psi^T H \Psi \, dt \right] = \begin{bmatrix} k_t K & c_t I_n \\ d_t I_n & m_t M^{-1} \end{bmatrix},$$

$$\boldsymbol{b}_{\mathcal{S}} = \int_{\mathcal{I}} \Psi^T \left( \boldsymbol{f}_z - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1} \right) \, dt,$$

and (see Appendix A for the explicit form of the time operators):

$$\begin{aligned} k_t &= \left( \int_{\mathcal{I}} \psi_q^2 \, dt \right), \\ c_t &= \left( \int_{\mathcal{I}} \psi_q \dot{\psi}_p \, dt \right), \\ d_t &= - \left( \int_{\mathcal{I}} \dot{\psi}_q \psi_p \, dt \right) = c_t - \psi_q(T)\psi_p(T), \\ m_t &= \left( \int_{\mathcal{I}} \psi_p^2 \, dt \right). \end{aligned}$$

The operator $M^{-1}$ is not computed explicitly. Instead, the Schur complement of $M^{-1}$ in $A_{\mathcal{S}}$ is considered. Equation (3.22) can thus be expanded as:

$$\begin{aligned} k_t K\ \boldsymbol{\varphi}_q \ &+ \qquad\quad c_t\ \boldsymbol{\varphi}_p = \boldsymbol{b}_q, \\ d_t\ \boldsymbol{\varphi}_q \ &+ \ \ m_t M^{-1}\ \boldsymbol{\varphi}_p = \boldsymbol{b}_p, \end{aligned}$$

so that:

$$\left[m_t k_t K - c_t d_t M\right] \boldsymbol{\varphi}_q = m_t \boldsymbol{b}_q - c_t M \boldsymbol{b}_p, \tag{3.23}$$

$$\boldsymbol{\varphi}_p = \frac{1}{m_t} M \left[\boldsymbol{b}_p - d_t \boldsymbol{\varphi}_q\right]. \tag{3.24}$$

Therefore, the solution of (3.22) amounts to solving (3.23) for $\boldsymbol{\varphi}_q$ by factorization of the sparse symmetric matrix:

$$A_q = m_t k_t K - c_t d_t M, \tag{3.25}$$

and inserting the solution $\boldsymbol{\varphi}_q$ into (3.24) to determine $\boldsymbol{\varphi}_p$.

For a given $m^{\text{th}}$ enrichment, the spatial modes $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ are subsequently projected to ensure that any new mode is searched in a direction that is orthogonal to the subspaces generated by the previous modes, respectively $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m-1}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m-1}$. At any given $m$, we want $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m}$ to be orthogonal with respect to $K$ and $M^{-1}$, respectively. Let $S_q$ and $S_p$ be defined as:

$$\begin{aligned} S_q &= \left[\boldsymbol{\varphi}_1^q \quad \ldots \quad \boldsymbol{\varphi}_{m-1}^q\right], \\ S_p &= \left[\boldsymbol{\varphi}_1^p \quad \ldots \quad \boldsymbol{\varphi}_{m-1}^p\right]. \end{aligned}$$

A classical approach consists in using the orthogonal projections:

$$\begin{aligned} P_q &= I_n - S_q \left(S_q^T K S_q\right)^{-1} S_q^T K, \\ P_p &= I_n - S_p \left(S_p^T M^{-1} S_p\right)^{-1} S_p^T M^{-1}. \end{aligned}$$

At any enrichment step, the previous modes $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m-1}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m-1}$ are orthogonal and normalized with respect to $K$ and $M^{-1}$, respectively. Thus, the projectors above simplify as:

$$\begin{aligned} P_q &= I_n - S_q S_q^T K, \\ P_p &= I_n - S_p S_p^T M^{-1}. \end{aligned}$$

Therefore, if we denote by $\boldsymbol{\varphi}_q^\circ$ and $\boldsymbol{\varphi}_p^\circ$ the modes initially obtained from Eqs. (3.23) and (3.24) and by $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ the modes that one retains after orthonormalization, the procedure reads:

$$\begin{aligned}
\boldsymbol{\varphi}_q^{\perp} &= P_q \boldsymbol{\varphi}_q^{\circ}, & \boldsymbol{\varphi}_p^{\perp} &= P_p \boldsymbol{\varphi}_p^{\circ}, \\
\boldsymbol{\varphi}_q &= \frac{\boldsymbol{\varphi}_q^{\perp}}{\sqrt{{\boldsymbol{\varphi}_q^{\perp}}^T K \boldsymbol{\varphi}_q^{\perp}}}, & \boldsymbol{\varphi}_p &= \frac{\boldsymbol{\varphi}_p^{\perp}}{\sqrt{{\boldsymbol{\varphi}_p^{\perp}}^T M^{-1} \boldsymbol{\varphi}_p^{\perp}}}.
\end{aligned}$$

It it noteworthy that, in practice, the inverse of $M$ is never evaluated. Instead, one performs a Cholesky factorization to obtain the decomposition $M = LL^T$. In particular, the normalization of $\boldsymbol{\varphi}_p$ is done by forward and backward substitution, whose cost is negligible with respect to the overall complexity of the algorithm. Indeed, the main bottleneck is the factorization of $A_q$ (3.25), which needs to be performed at each iteration of the fixed point algorithm. We propose below two approaches that aim at:

- Reducing the number of iterations in the fixed point algorithm in order to reach convergence (see section 3.3.2);
- Avoiding repetitive factorization of $A_q$ by carrying out computations in a subspace provided by the Ritz vectors, which are approximations of the eigenvectors of the generalized eigenproblem $K\boldsymbol{u} = \lambda M \boldsymbol{u}$ (see Section 3.3.4).

### Problem in time

We assume here that $\boldsymbol{\varphi}$ is known and search for a new temporal mode $\boldsymbol{\psi}$. We substitute $\boldsymbol{z}_{m-1} + \Phi\boldsymbol{\psi}$ for $\boldsymbol{z}$ in (3.18) and choose test functions in the form $\boldsymbol{z}^* = \Phi\boldsymbol{\psi}^*$, with $\boldsymbol{\psi}^* \in \mathcal{Y}^2$, where:

$$\mathcal{Y} = C^0(\bar{\mathcal{I}}).$$

Equation (3.18) reduces in this case to:

$$\int_{\mathcal{I}} \boldsymbol{\psi}^{*T} \Phi^T \left( J_{2n} \Phi \dot{\boldsymbol{\psi}} + H \Phi \boldsymbol{\psi} \right) \, dt = \int_{\mathcal{I}} \boldsymbol{\psi}^{*T} \Phi^T \left( \boldsymbol{f}_z - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1} \right) \, dt, \qquad \forall \boldsymbol{\psi}^* \in \mathcal{Y}^2,$$

which simplifies to:

$$\Phi^T J_{2n} \Phi \dot{\boldsymbol{\psi}} + \Phi^T H \Phi \boldsymbol{\psi} = \Phi^T \left( \boldsymbol{f}_z - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1} \right).$$

Above equation is discretized using the Crank-Nicolson time-marching scheme, such that, given $\boldsymbol{\psi}^0$, one computes the $i^{\text{th}}$ iterate ($i > 0$) as:

$$A_{\mathcal{T}} \boldsymbol{\psi}^i = B_{\mathcal{T}} \boldsymbol{\psi}^{i-1} + \boldsymbol{b}_{\mathcal{T}}^i, \qquad i = 1, \ldots, n_t \tag{3.26}$$

where:

$$A_{\mathcal{T}} = \begin{bmatrix} h_t k_x & 2c_x \\ -2c_x & h_t m_x \end{bmatrix},$$

$$B_{\mathcal{T}} = \begin{bmatrix} -h_t k_x & 2c_x \\ -2c_x & -h_t m_x \end{bmatrix},$$

$$\boldsymbol{b}_{\mathcal{T}}^i = \Phi^T \Big[ h_t \Big( \boldsymbol{f}_z^i + \boldsymbol{f}_z^{i-1} - H \left( \boldsymbol{z}_{m-1}^i + \boldsymbol{z}_{m-1}^{i-1} \right) \Big) - 2J_{2n} \Big( \boldsymbol{z}_{m-1}^i - \boldsymbol{z}_{m-1}^{i-1} \Big) \Big],$$

and:

$$\begin{aligned} k_x &= \boldsymbol{\varphi}_q^T K \boldsymbol{\varphi}_q, \\ c_x &= \boldsymbol{\varphi}_q^T \boldsymbol{\varphi}_p, \\ m_x &= \boldsymbol{\varphi}_p^T M^{-1} \boldsymbol{\varphi}_p. \end{aligned}$$

For each time step, Eq. (3.26) represents a $2 \times 2$ linear system that can be explicitly solved for $\boldsymbol{\psi}^n$. Overall, the time problem is relatively cheap to solve as the cost is linear in the number of time steps $n_t$. As previously mentioned, $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ are normalized after (3.22) is solved, so that $k_x = m_x = 1$ and only $c_x$ needs to be updated.

### 3.3.2 Aitken acceleration

In the context of PGD order-reduced modeling, the number of iterations performed by the fixed-point algorithm has a direct impact on the efficiency of the approach. We propose here to employ the Aitken's $\Delta^2$ process to reduce the number of iterations that are necessary to reach convergence.

Let $\text{lin}(n)$ denote the complexity associated with solving one linear system of $n$ algebraic equations in $n$ unknown variables ($\text{lin}(n) \approx \mathcal{O}(n^3)$ for fully-populated matrices). In structural dynamics simulations, the usual approach is to discretize the continuous equations with respect to the spatial variables using the finite element method and then obtain a system of $n$ ordinary differential equations in the time variable $t \in \mathcal{I}$. The system is thereafter discretized in time by means of an (implicit) integration scheme (e.g. Euler, Newmark, Crank-Nicolson, Hilber-Hughes-Taylor, ...). The degrees of freedom are then evaluated at each time step by solving a linear system of size $n$. In the case of $n_t$ time steps, the complexity of the approach amounts to $n_t\text{lin}(n)$.

In the PGD framework, the solution of the problems in space and time is decoupled such that at each fixed-point iteration, one system of size $n$ is solved for the spatial mode (3.23) and one system of size two is solved $n_t$ times (marching scheme) for the temporal mode (3.26). The complexity of one fixed-point iteration can be assumed to be of the order of $\mathrm{lin}(n) + n_t$. It follows that the overall complexity of the PGD algorithm will be $mk_{\max}(\mathrm{lin}(n) + n_t)$, where $m$ denotes the rank of the decomposition, i.e. the number of modes, and $k_{\max}$ is the maximal number of iterations allowed in the fixed-algorithm, whether or not convergence is reached. It can be inferred that a space-time separated PGD algorithm is competitive against a classical full-order solver whenever the following inequality holds:

$$mk_{\max}(\mathrm{lin}(n) + n_t) \ll n_t \mathrm{lin}(n),$$

highlighting the fact that the efficiency of the PGD algorithm highly depends on the number of fixed-point iterations.

The computation of an enrichment mode involves the following operators, formally written, at any given fixed-point iteration $k$:

- $\mathcal{S}^{(k)} : \boldsymbol{\psi}^{(k-1)} \mapsto \boldsymbol{\varphi}^{(k)}$, the operator that solves the system (3.22) for $\boldsymbol{\varphi}^{(k)}$ with $\boldsymbol{\psi}^{(k-1)}$ given;
- $\mathcal{T}^{(k)} : \boldsymbol{\varphi}^{(k)} \mapsto \boldsymbol{\psi}^{(k)}$, the operator that solves the system (3.26) for $\boldsymbol{\psi}^{(k)}$ with $\boldsymbol{\varphi}^{(k)}$ given.

As a result, the fixed-point algorithm computes two sequences for spatial and temporal modes, i.e. $\left(\boldsymbol{\varphi}^{(k)}\right)_{1\leqslant k\leqslant k_{\max}}$ and $\left(\boldsymbol{\psi}^{(k)}\right)_{1\leqslant k\leqslant k_{\max}}$, until convergence. These sequences can be defined by recurrence relations as follows:

$$\begin{aligned}
\boldsymbol{\varphi}^{(k)} &= \mathcal{S}^{(k)} \circ \mathcal{T}^{(k-1)}\left(\boldsymbol{\varphi}^{(k-1)}\right),\\
\boldsymbol{\psi}^{(k)} &= \mathcal{T}^{(k)} \circ \mathcal{S}^{(k)}\left(\boldsymbol{\psi}^{(k-1)}\right).
\end{aligned}$$

The fixed-point convergence hinges upon the contraction property of the operators $\mathcal{S}^{(k)} \circ \mathcal{T}^{(k-1)}$ and $\mathcal{T}^{(k)} \circ \mathcal{S}^{(k)}$ for $\boldsymbol{\varphi}^{(k)}$ and $\boldsymbol{\psi}^{(k)}$ respectively. One common way to improve fixed-point iterations is by using relaxation techniques. This helps achieve a contraction property and usually enhances the convergence rate. The introduction of relaxation parameters $\omega_\varphi$ and $\omega_\psi$ leads to the following formulation of a fixed-point iteration:

$$\begin{aligned}
\boldsymbol{\varphi}^{(k)} &= \omega_\varphi\ \mathcal{S}^{(k)} \circ \mathcal{T}^{(k-1)}\left(\boldsymbol{\varphi}^{(k-1)}\right) + (1-\omega_\varphi)\,\boldsymbol{\varphi}^{(k-1)},\\
\boldsymbol{\psi}^{(k)} &= \omega_\psi\ \mathcal{T}^{(k)} \circ \mathcal{S}^{(k)}\left(\boldsymbol{\psi}^{(k-1)}\right) \quad + (1-\omega_\psi)\,\boldsymbol{\psi}^{(k-1)}.
\end{aligned}$$

In practice, it is preferable to adapt $\omega_\varphi$ and $\omega_\psi$ at each iteration. The so-called Aitken's delta square method provides a useful heuristic for determining the sequences $\omega_\varphi^{(k)}$ and $\omega_\psi^{(k)}$. One can also choose to enforce relaxation on the generalized coordinates modes and the conjugate momentum modes separately. In the Algorithm 3, Aitken acceleration is applied on the spatial modes only and separately for $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$. Note that steps 15 and 16 of Algorithm 3 are not implemented in practice. Instead, space-time separation should be leveraged to efficiently compute stagnation coefficients in step 16.

**Algorithm 3** Fixed point algorithm with Aitken acceleration

```
1: **Initialization:** Set $k \leftarrow 0$
2:                 Set $s_q \leftarrow \epsilon + 1$ and $s_p \leftarrow \epsilon + 1$ (with $\epsilon = 10^{-9}$)
3:                 Set $\boldsymbol{\psi}^{(0)}$ and $\boldsymbol{\varphi}^{(0)}$
4: **while** $k < k_{\max}$ and $\max(s_q, s_p) > \epsilon$ **do**
5:     Increment the iteration counter: $k \leftarrow k + 1$
6:     Compute new spatial modes: $\boldsymbol{\varphi} \leftarrow \mathcal{S}^{(k)}\left(\boldsymbol{\psi}^{(k-1)}\right)$
7:     Project modes: $\boldsymbol{\varphi}_q \leftarrow P_q\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p \leftarrow P_p\boldsymbol{\varphi}_p$
8:     Normalize modes: $\boldsymbol{\varphi}_q^{(k)} \leftarrow \frac{\boldsymbol{\varphi}_q}{\sqrt{\boldsymbol{\varphi}_q^T K \boldsymbol{\varphi}_q}}$ and $\boldsymbol{\varphi}_p^{(k)} \leftarrow \frac{\boldsymbol{\varphi}_p}{\sqrt{\boldsymbol{\varphi}_p^T M^{-1} \boldsymbol{\varphi}_p}}$
9:     Update spatial residual: $\boldsymbol{r}_q^{(k)} = \boldsymbol{\varphi}_q^{(k)} - \boldsymbol{\varphi}_q^{(k-1)}$ and $\boldsymbol{r}_p^{(k)} = \boldsymbol{\varphi}_p^{(k)} - \boldsymbol{\varphi}_p^{(k-1)}$
10:    **if** $k > 1$ **then**
11:        Aitken $\Delta^2$:
               $\boldsymbol{\varphi}_q^{(k)} \leftarrow \omega_q^{(k)}\boldsymbol{\varphi}_q^{(k)} + \left(1 - \omega_q^{(k)}\right)\boldsymbol{\varphi}_q^{(k-1)}$ with $\omega_q^{(k)} = \omega_q^{(k-1)} \frac{\boldsymbol{r}_q^{(k-1)^T}\left(\boldsymbol{r}_q^{(k)} - \boldsymbol{r}_q^{(k-1)}\right)}{\left\|\boldsymbol{r}_q^{(k)} - \boldsymbol{r}_q^{(k-1)}\right\|^2}$
               $\boldsymbol{\varphi}_p^{(k)} \leftarrow \omega_p^{(k)}\boldsymbol{\varphi}_p^{(k)} + \left(1 - \omega_p^{(k)}\right)\boldsymbol{\varphi}_p^{(k-1)}$ with $\omega_p^{(k)} = \omega_p^{(k-1)} \frac{\boldsymbol{r}_p^{(k-1)^T}\left(\boldsymbol{r}_p^{(k)} - \boldsymbol{r}_p^{(k-1)}\right)}{\left\|\boldsymbol{r}_p^{(k)} - \boldsymbol{r}_p^{(k-1)}\right\|^2}$
12:    **end if**
13:    Compute new temporal mode: $\boldsymbol{\psi}^{(k)} \leftarrow \mathcal{T}^{(k)}\left(\boldsymbol{\varphi}^{(k)}\right)$
14:    Compute: $\Delta_q \leftarrow \boldsymbol{\varphi}_q^{(k)}\boldsymbol{\psi}_q^{(k)} - \boldsymbol{\varphi}_q^{(k-1)}\boldsymbol{\psi}_q^{(k-1)}$ and $\Sigma_q \leftarrow \frac{1}{2}\left(\boldsymbol{\varphi}_q^{(k)}\boldsymbol{\psi}_q^{(k)} + \boldsymbol{\varphi}_q^{(k-1)}\boldsymbol{\psi}_q^{(k-1)}\right)$
15:             $\Delta_p \leftarrow \boldsymbol{\varphi}_p^{(k)}\boldsymbol{\psi}_p^{(k)} - \boldsymbol{\varphi}_p^{(k-1)}\boldsymbol{\psi}_p^{(k-1)}$ and $\Sigma_p \leftarrow \frac{1}{2}\left(\boldsymbol{\varphi}_p^{(k)}\boldsymbol{\psi}_p^{(k)} + \boldsymbol{\varphi}_p^{(k-1)}\boldsymbol{\psi}_p^{(k-1)}\right)$
16:    Evaluate the stagnation coefficients: $s_q \leftarrow \|\Delta_q\|_{L^2}/\|\Sigma_q\|_{L^2}$ and $s_p \leftarrow \|\Delta_p\|_{L^2}/\|\Sigma_p\|_{L^2}$
17: **end while**
18: **Return** the modes $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi}^{(k)}$ and $\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi}^{(k)}$
```

### 3.3.3 Temporal update and symplectic structure

Greedy algorithms generally incorporate an update procedure that consists in updating all the temporal modes for a given set of spatial modes. For a decomposition of rank $m$, the spatial modes can be conveniently stored in the matrix $S$ of size $2n \times 2m$, defined as:

$$S = \begin{bmatrix} \boldsymbol{\varphi}_1^q & \dots & \boldsymbol{\varphi}_m^q & & 0 & \\ & 0 & & \boldsymbol{\varphi}_1^p & \dots & \boldsymbol{\varphi}_m^p \end{bmatrix} = \begin{bmatrix} S_q & 0 \\ 0 & S_p \end{bmatrix},$$

while the temporal modes can be vertically stored in the time-dependent vector $\boldsymbol{\psi}$ of size $2m \times 1$, defined as:

$$\boldsymbol{\psi} = \begin{bmatrix} \psi_1^q \\ \vdots \\ \psi_m^q \\ \psi_1^p \\ \vdots \\ \psi_m^p \end{bmatrix},$$

such that the decomposition of rank $m$ of $\boldsymbol{z}$ reads:

$$\boldsymbol{z}_m(t) = S\boldsymbol{\psi}(t).$$

The temporal update is performed by substituting $S\boldsymbol{\psi}$ for $\boldsymbol{z}$ in (3.18) and choosing test functions in the form $\boldsymbol{z}^* = S\boldsymbol{\psi}^*$. Equation (3.18) thus reduces to:

$$\int_{\mathcal{I}} \boldsymbol{\psi}^{*T} S^T \left( J_{2n} S\dot{\boldsymbol{\psi}} + HS\boldsymbol{\psi} \right) \, dt = \int_{\mathcal{I}} \boldsymbol{\psi}^{*T} S^T \boldsymbol{f}_z \, dt, \qquad \forall \boldsymbol{\psi}^* \in \mathcal{Y}^{2m},$$

which can be rewritten in matrix form, with $\boldsymbol{f}_\psi = S^T \boldsymbol{f}_z$, as:

$$S^T J_{2n} S\dot{\boldsymbol{\psi}} + S^T HS\boldsymbol{\psi} = \boldsymbol{f}_\psi. \tag{3.27}$$

Time discretization of the above equation using the Crank-Nicolson marching scheme leads to:

$$A_{\mathcal{U}}\boldsymbol{\psi}^i = B_{\mathcal{U}}\boldsymbol{\psi}^{i-1} + \boldsymbol{b}_{\mathcal{U}}^i, \qquad i = 1, \dots, n_t, \tag{3.28}$$

with:

$$A_{\mathcal{U}} = \begin{bmatrix} h_t K_x & 2C_x \\ -2C_x^T & h_t M_x \end{bmatrix}, \qquad B_{\mathcal{U}} = \begin{bmatrix} -h_t K_x & 2C_x \\ -2C_x^T & -h_t M_x \end{bmatrix},$$

$$\boldsymbol{b}_{\mathcal{U}}^i = h_t S^T \left( \boldsymbol{f}_z^i + \boldsymbol{f}_z^{i-1} \right),$$

and:

$$K_x = S_q^T K S_q,$$
$$C_x = S_q^T S_p,$$
$$M_x = S_p^T M^{-1} S_p.$$

The orthonormalization of $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m}$ with $K$ and $M^{-1}$, respectively, results in $K_x = M_x = I_m$.

The update procedure can be interpreted as projecting Hamilton's equations onto the subspace generated by the vectors of $S$. The system to be solved is governed by the Hamiltonian $g$ whose Hessian is $G = S^T H S$. This Hessian can be interpreted as the rank-$2m$ reduced counterpart of the Hessian operator $H$, such that:

$$g(\boldsymbol{\psi}) = \frac{1}{2} \boldsymbol{\psi}^T G \boldsymbol{\psi} - \boldsymbol{\psi}^T \boldsymbol{f}_\psi,$$

and the full-order vector is given by $\boldsymbol{z} \simeq \boldsymbol{z}_m = S\boldsymbol{\psi}$. Assuming that the Hamiltonian $g$ is canonical, the Hamilton's canonical equations of such a reduced-order system read:

$$\dot{\boldsymbol{\psi}} = \nabla^\gamma g,$$

where the symplectic gradient is given by:

$$\nabla^\gamma g = J_{2m} \nabla g = J_{2m} \left( G\boldsymbol{\psi} + \boldsymbol{f}_\psi \right).$$

It follows that the Hamilton's equations can be written as:

$$\dot{\boldsymbol{\psi}} = J_{2m} \left( G\boldsymbol{\psi} - \boldsymbol{f}_\psi \right).$$

Multiplying both sides of this equation by $J_{2m}$ (recall that $J_{2m} J_{2m} = -I_{2m}$) and rearranging the terms leads to:

$$J_{2m} \dot{\boldsymbol{\psi}} + S^T H S \boldsymbol{\psi} = \boldsymbol{f}_\psi.$$

Recalling here Eq. (3.27):

$$S^T J_{2n} S \dot{\boldsymbol{\psi}} + S^T H S \boldsymbol{\psi} = \boldsymbol{f}_\psi,$$

one observes that the two equations are identical if and only if $S^T J_{2n} S = J_{2m}$, i.e. if $S$ is a symplectic mapping, according to the definition of the symplectic Stiefel manifold (3.19). However, in general, $S$ is not symplectic nor $g$ is a canonical Hamiltonian. The product $S^T J_{2n} S$ writes:

$$S^T J_{2n} S = \begin{bmatrix} 0 & S_q^T S_p \\ -S_p^T S_q & 0 \end{bmatrix}.$$

This suggests that the symplectic property could be enforced by biorthogonalization of $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m}$, such that:

$$S^T J_{2n} S = \begin{bmatrix} 0 & I_m \\ -I_m & 0 \end{bmatrix} = J_{2m}.$$

However, this property is not ensured in the current algorithm since we chose to orthogonalize $(\boldsymbol{\varphi}_i^q)_{1\leqslant i\leqslant m}$ and $(\boldsymbol{\varphi}_i^p)_{1\leqslant i\leqslant m}$ with respect to $K$ and $M^{-1}$, respectively. Yet, it can be enforced via a post-processing procedure. Let $P$ and $Q$ be two matrices of size $m \times m$ such that:

$$\hat{S}_q^T \hat{S}_p = I_m, \qquad \text{with } \hat{S}_q = S_q Q, \text{ and } \hat{S}_p = S_p P.$$

It follows that:

$$(S_q Q)^T S_p P = Q^T S_q^T S_p P = I_m \tag{3.29}$$

In other words, the matrices $Q$ and $P$ recombine the columns of $S_q$ and $S_p$ such that $(\hat{\boldsymbol{\varphi}}_i^q)_{1\leqslant i\leqslant m}$ and $(\hat{\boldsymbol{\varphi}}_i^p)_{1\leqslant i\leqslant m}$ form a biorthogonal system. We can readily conceive two approaches, among others, to enforce (3.29):

- The LU factorization $S_q^T S_p = LU$ allows one to define $Q = L^{-T}$ and $P = U^{-1}$;
- The Singular Value Decomposition $S_q^T S_p = U\Sigma V^T$ allows one to define $Q = U^{-T}\Sigma^{-1/2}$ and $P = V^{-T}\Sigma^{-1/2}$ ($\Sigma^{-1/2}$ is defined as the diagonal matrix whose coefficients are given by the square root of the inverse of the singular values if different from zero, and zero otherwise).

We note that the two procedures are computationally efficient since they are performed on reduced matrices ($m \ll n$). Therefore, it is possible to construct a symplectic basis by post-processing the basis calculated by the PGD solver.

> **Remark** Instead of post-processing the spatial bases, symplecticity can be enforced throughout the computation. The transformation matrix $S$ can be made symplectic by

considering the two following projections:

$$P_q = I_n - S_p S_q^T,$$
$$P_p = I_n - S_q S_p^T,$$

instead of those proposed in 3.3.1. Furthermore, if we denote by $\boldsymbol{\varphi}_q^\circ$ and $\boldsymbol{\varphi}_p^\circ$ the modes initially obtained from Eqs. (3.23) and (3.24) and by $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ the modes that one retains after biorthonormalization, the procedure reads:

$$\boldsymbol{\varphi}_q^\perp = P_q \boldsymbol{\varphi}_q^\circ, \qquad \boldsymbol{\varphi}_p^\perp = P_p \boldsymbol{\varphi}_p^\circ,$$
$$\boldsymbol{\varphi}_q = \frac{\boldsymbol{\varphi}_q^\perp}{\|\boldsymbol{\varphi}_q^\perp\|^2}, \qquad \boldsymbol{\varphi}_p = \frac{\boldsymbol{\varphi}_p^\perp}{\boldsymbol{\varphi}_p^{\perp T} \boldsymbol{\varphi}_q},$$

with $\|\cdot\|$ the Euclidean norm. Then, after convergence of the fixed-point iteration, biorthogonalization can be checked using the following modified Gram-Schmidt algorithm:

$$\begin{aligned} \boldsymbol{\varphi}_q &\leftarrow \boldsymbol{\varphi}_q - \left(\boldsymbol{\varphi}_q^T \boldsymbol{\varphi}_j^p\right) \boldsymbol{\varphi}_j^q, \\ \boldsymbol{\varphi}_p &\leftarrow \boldsymbol{\varphi}_p - \left(\boldsymbol{\varphi}_p^T \boldsymbol{\varphi}_j^q\right) \boldsymbol{\varphi}_j^p, \end{aligned} \qquad j = 1, \ldots, m-1$$

$$\boldsymbol{\varphi}_q \leftarrow \frac{\boldsymbol{\varphi}_q}{\|\boldsymbol{\varphi}_q\|^2},$$
$$\boldsymbol{\varphi}_p \leftarrow \frac{\boldsymbol{\varphi}_p}{\boldsymbol{\varphi}_p^T \boldsymbol{\varphi}_q}$$

While this procedure ensures that $S$ is symplectic, the orthonormality of $S_q$ and $S_p$ with respect to $K$ and $M^{-1}$, respectively, is removed. In other words, the relation $C_x = I_m$ is enforced but $K_x \neq I_m$ and $M_x \neq I_m$.

### 3.3.4 Projection in Ritz subspace

As previously mentioned, the main bottleneck of the PGD solver is the solution of (3.22) that requires one to factorize the operator $A_q$ at each fixed-point iteration. Even though Aitken transformation does reduce the PGD solver time, the computational cost of the repeated factorization makes the solver prohibitively expensive when the dimension of the finite element space is large.

We recall here the problem in space (3.22), expressed now at a given fixed-point iteration indexed by parameter $k$:

$$A_S^{(k)} \boldsymbol{\varphi}^{(k)} = \boldsymbol{b}_S^{(k)}$$

with:

$$A_{\mathcal{S}}^{(k)} = \left[\int_{\mathcal{I}} \Psi^{(k-1)^T} J_{2n} \dot{\Psi}^{(k-1)} + \Psi^{(k-1)^T} H \Psi^{(k-1)} \; dt\right] = \begin{bmatrix} k_t^{(k)} K & c_t^{(k)} I_n \\ d_t^{(k)} I_n & m_t^{(k)} M^{-1} \end{bmatrix}$$

$$\boldsymbol{b}_{\mathcal{S}}^{(k)} = \int_{\mathcal{I}} \Psi^{(k-1)^T} \left(\boldsymbol{f}_z - J_{2n} \dot{\boldsymbol{z}}_{m-1} - H \boldsymbol{z}_{m-1}\right) \; dt$$

where $m_t^{(k)}$, $k_t^{(k)}$, $c_t^{(k)}$ and $d_t^{(k)}$ are computed from the temporal modes $\psi_q^{(k-1)}$ and $\psi_p^{(k-1)}$, as defined in Section 3.3.1. In particular:

$$A_q^{(k)} = \left[m_t^{(k)} k_t^{(k)} K - c_t^{(k)} d_t^{(k)} M\right]$$

Therefore, at each fixed-point iteration, the weights associated with the stiffness and mass operators $K$ and $M$, respectively, have to be modified and a new factorization of $A_q^{(k)}$ needs to be obtained.

Although $A_q^{(k)}$ varies from one iteration to the next, its spectral content remains similar because the operator is derived from a linear combination of $K$ and $M$ (both remaining constant). The proposed approach takes advantage of the later observation and consists in projecting Eq. (3.23) onto the subspace of approximated eigenvectors, namely the Ritz vectors, which verify the following properties (with $m \leqslant r \ll n$):

$$\left(\hat{\Lambda}, \hat{V}\right) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{n\times r}, \quad \text{such that} \quad \hat{V}^T K \hat{V} = \hat{\Lambda}, \text{ and } \quad \hat{V}^T M \hat{V} = I_r,$$

where the Ritz values and the Ritz vectors are:

$$\hat{\Lambda} = \operatorname{diag}\left(\hat{\lambda}_1, \; \dots \; , \hat{\lambda}_r\right),$$
$$\hat{V} = \left[\hat{\boldsymbol{v}}_1 \; \dots \; \hat{\boldsymbol{v}}_r\right].$$

We now introduce the mapping $R$:

$$R = \begin{bmatrix} \hat{V} & 0 \\ 0 & M\hat{V} \end{bmatrix},$$

and remark that $R \in S_p(2r, 2n)$, i.e. $R$ is a symplectic mapping. In other words, the structure of the equations presented above holds, which can be written in terms of $\hat{\boldsymbol{z}} \in \mathbb{R}^r$, that satisfies $\boldsymbol{z} = R\hat{\boldsymbol{z}}$, and the Hamiltonian $g$ defined as:

$$g(\hat{\boldsymbol{z}}) = \frac{1}{2}\hat{\boldsymbol{z}}^T G \hat{\boldsymbol{z}} - \hat{\boldsymbol{z}}^T \boldsymbol{f}_{\hat{z}},$$

with:

$$G = R^T H R = \begin{bmatrix} \hat{\Lambda} & 0 \\ 0 & I_r \end{bmatrix}, \qquad \boldsymbol{f}_{\hat{z}} = R^T \boldsymbol{f}_z.$$

For the Hamiltonian $g$, the problem in space (3.22) using $\boldsymbol{\varphi}^{(k)} = R\hat{\boldsymbol{\varphi}}^{(k)}$ can thus be rewritten as:

$$\hat{A}_{\mathcal{S}}^{(k)} \hat{\boldsymbol{\varphi}}^{(k)} = \hat{\boldsymbol{b}}_{\mathcal{S}}^{(k)}, \tag{3.30}$$

with:

$$\hat{A}_{\mathcal{S}}^{(k)} = R^T A_{\mathcal{S}}^{(k)} R = \left[ \int_{\mathcal{I}} \Psi^{(k-1)^T} J_{2r} \dot{\Psi}^{(k-1)} + \Psi^{(k-1)^T} G \Psi^{(k-1)} \, dt \right] = \begin{bmatrix} k_t^{(k)} \hat{\Lambda} & c_t^{(k)} I_r \\ d_t^{(k)} I_r & m_t^{(k)} I_r \end{bmatrix},$$

$$\hat{\boldsymbol{b}}_{\mathcal{S}}^{(k)} = R^T \boldsymbol{b}_{\mathcal{S}}^{(k)},$$

and (3.23) becomes a diagonal system expressed as:

$$\left[ m_t^{(k)} k_t^{(k)} \hat{\Lambda} - c_t^{(k)} d_t^{(k)} I_r \right] \hat{\boldsymbol{\varphi}}_q^{(k)} = \hat{\boldsymbol{b}}_q^{(k)}. \tag{3.31}$$

The complexity of the spatial problem (3.22) is now linear in terms of the dimension $r$ of the Ritz subspace. The number of Ritz vectors $r$ has to be chosen sufficiently high with respect to the expected rank $m$ of the PGD approximation. Depending on the external load, one can compute the Ritz vectors associated to the Ritz values corresponding to the frequency band of interest. Here, we chose to retain the Ritz vectors whose Ritz values have the lowest magnitudes, as conventionally performed in structural dynamics [15].

## 3.4 Numerical examples and discussion

### 3.4.1 Test case: asymmetric triangle wave Neumann boundary condition

The test case is inspired by an example found in [29] and has the interest of showcasing a transient phase followed by a steady-state harmonic regime. A 3D beam is considered, such that the domain $\Omega = (0, 6) \times (0, 1) \times (0, 1)$ (in meters) is a parallelepiped with a squared cross-section (see Figure 3.1). Its response to an external load on its top surface is computed in $\mathcal{I} = (0, 5)$ (in seconds):

$$\rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma(u) = 0, \qquad \forall (x, t) \in \Omega \times \mathcal{I},$$

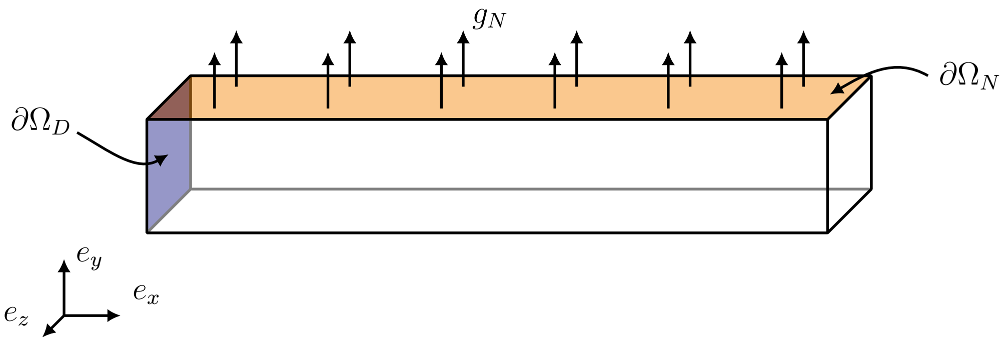


Figure 3.1 Scheme of the test case.

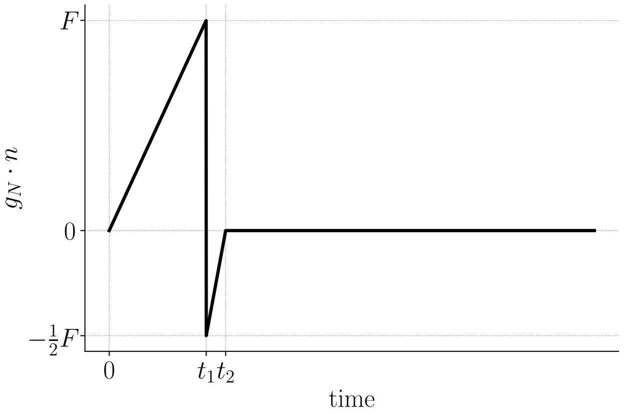


Figure 3.2 Evolution in time of the boundary traction $g_N \cdot n$.

with:

$$\begin{aligned} \sigma(u) &= 2\mu\varepsilon(u) + \lambda \mathrm{tr}\left(\varepsilon(u)\right) I_3, \\ \varepsilon(u) &= \frac{1}{2}\left(\nabla u + (\nabla u)^T\right). \end{aligned}$$

Moreover, the beam is subjected to homogeneous initial conditions:

$$\begin{aligned} u(x,0) &= 0, && \forall x \in \Omega, \\ \frac{\partial u}{\partial t}(x,0) &= 0, && \forall x \in \Omega, \end{aligned}$$

and to the boundary conditions:

$$\begin{aligned} u(x,t) &= 0, && \forall (x,t) \in \partial\Omega_D \times \mathcal{I}, \\ \sigma(u)\cdot n &= g_N(x,t), && \forall (x,t) \in \partial\Omega_N \times \mathcal{I}, \\ \sigma(u)\cdot n &= 0, && \forall (x,t) \in \partial\Omega_0 \times \mathcal{I}. \end{aligned}$$

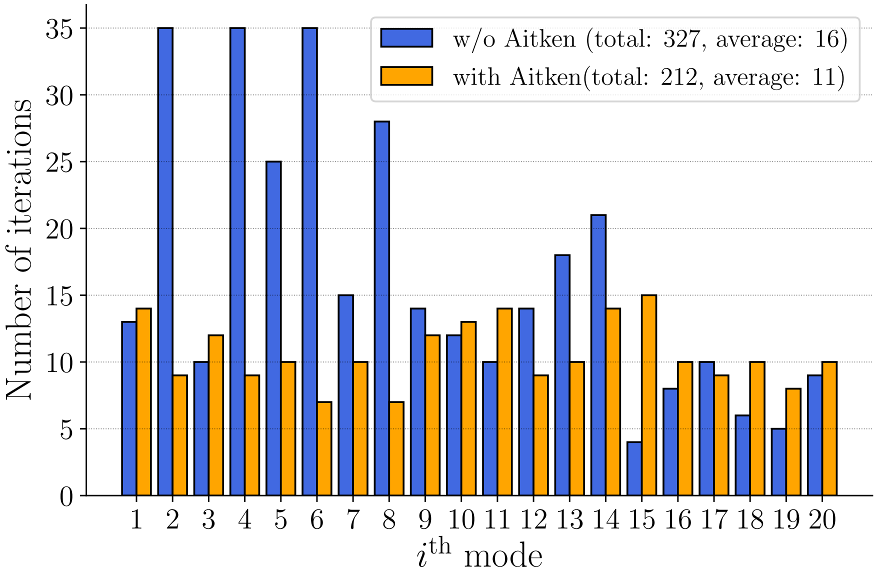


Figure 3.3 Number of iterations for 20 modes without and with Aitken acceleration

In other words, the beam is clamped on its left end $\partial\Omega_D = \{0\} \times (0,1) \times (0,1)$, an external load $g_N \cdot n$ is applied on its top surface $\partial\Omega_N = (0,6) \times \{1\} \times (0,1)$ such that:

$$g_N(x,t) \cdot n = \begin{cases} \dfrac{t}{t_1} F, & \text{if } t < t_1, \\ -\dfrac{1}{2}\left(1 - \dfrac{t - t_1}{t_2 - t_1}\right) F, & \text{if } t_1 \leqslant t < t_2, \\ 0, & \text{otherwise,} \end{cases}$$

where $t_1 = 0.625$ and $t_2 = 0.75$. In other words, the external load pulls the beam upwards for $t \in [0, t_1)$ and pushes it downwards for $t \in [t_1, t_2)$ (see Figure 3.2). Finally, the beam is free on the remainder of the boundary $\partial\Omega_0 = \partial\Omega \backslash (\partial\Omega_D \cup \partial\Omega_N)$. In the space-discrete, time-continuous Hamiltonian formalism, the problem reads:

$$\begin{aligned} \dot{\boldsymbol{p}} &= -K\boldsymbol{q} + \boldsymbol{f}, \\ \dot{\boldsymbol{q}} &= M^{-1}\boldsymbol{p}, \end{aligned}$$

with:

$$\begin{aligned} \boldsymbol{q}(0) &= 0, \\ \boldsymbol{p}(0) &= 0, \end{aligned}$$

where the stiffness and mass matrices, $K$ and $M$ respectively, result from the enforcement of the homogeneous Dirichlet boundary conditions by eliminating the corresponding rows

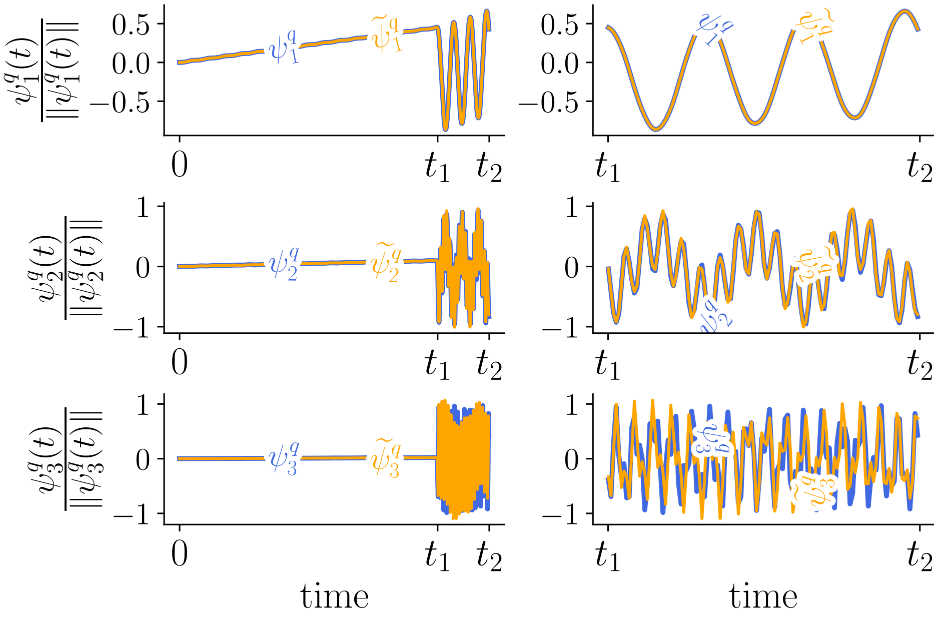


Figure 3.4 Visualization [95] of the first three temporal modes (normalized) with and without Aiken acceleration, herein denoted $\widetilde{\psi}_i^q$ and $\psi_i^q$, respectively.

and columns; the right-hand side $\boldsymbol{f}$ is computed from the prescribed Neumann boundary conditions.

The values of the parameters are chosen as follows:

$$\begin{aligned} E &= 220 \text{ GPa}, \\ \nu &= 0.3, \\ \rho &= 7000 \text{ kg/m}^3, \\ F &= 0.5 \text{ GPa}, \end{aligned}$$

and the Lamé coefficients are evaluated as:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

The time domain $\mathcal{I}$ is divided into $n_t = 4800$ sub-intervals of equal size. The space domain $\Omega$ is partitioned into linear tetrahedral elements and five discretizations will be considered such that the number of spatial DOFs $2n$ is chosen in $\{1{,}302, 6{,}204, 36{,}774, 67{,}032, 244{,}926\}$. The number of Ritz vectors is set to $r = 300$ regardless of the spatial discretization. Unless otherwise stated, the reduced-order models are assessed on solutions involving $m = 50$ modes.

### 3.4.2 Comparison method and performance criteria

We shall report the results based on the following four features:

1. The number of fixed-point iterations without and with Aitken acceleration;
2. The accuracy of the PGD approximations with respect to full-order solutions, namely the FEM solutions described in Section 3.1.3;
3. The actual execution time of the different approaches and algorithms. The time efficiency of the PGD solvers will be detailed regarding the successive phases of the computation, namely the pre-processing, the fixed-point algorithm, the Gram-Schmidt algorithm, and the temporal update procedure.
4. The scalability of the approaches with respect to the size of the spatial discretization.

The relative error $\epsilon_{ROM}$ in the reduced-order approximations with respect to the full-order solutions computed by the FEM is given by:

$$\epsilon_{ROM} = \frac{\vert\!\vert\!\vert u_{FEM} - u_{ROM} \vert\!\vert\!\vert}{\vert\!\vert\!\vert u_{FEM} \vert\!\vert\!\vert}$$

with $\vert\!\vert\!\vert \cdot \vert\!\vert\!\vert$ being the energy norm:

$$\vert\!\vert\!\vert u \vert\!\vert\!\vert = \sqrt{\int_{\mathcal{I}} \int_{\Omega} \frac{1}{2}\rho \dot{u} \cdot \dot{u} + \frac{1}{2}\epsilon(u) : \mathbb{E} : \epsilon(u) \; dxdt}.$$

More precisely, in the space-discrete Hamiltonian framework, the energy norm will be evaluated as follows:

$$\vert\!\vert\!\vert u \vert\!\vert\!\vert = \sqrt{\int_{\mathcal{I}} \frac{1}{2}\boldsymbol{p}^T M^{-1} \boldsymbol{p} + \frac{1}{2}\boldsymbol{q}^T K \boldsymbol{q} \; dt}.$$

Note that the full-order solution computed by the FEM is obtained using the same discretization parameters.

The reduced-order approximations that will be considered are the Singular Value Decomposition (SVD) of the full-order solution, the PGD LU that factorizes the space operator by LU decomposition for each fixed-point iteration and the PGD Ritz that computes the reduced-order model in the subspace spanned by the Ritz vectors. More precisely, we will present the errors with respect to the number of modes $m$ in the PGD solutions and compare these errors to those obtained by subsequently performing an SVD on the full-order solutions.

As far as computer times are concerned, all computations were run on a computer with the following configuration:

– CPU: AMD Ryzen 7 PRO 4750U @ 1.7 GHz per core (8 cores, 16 threads);

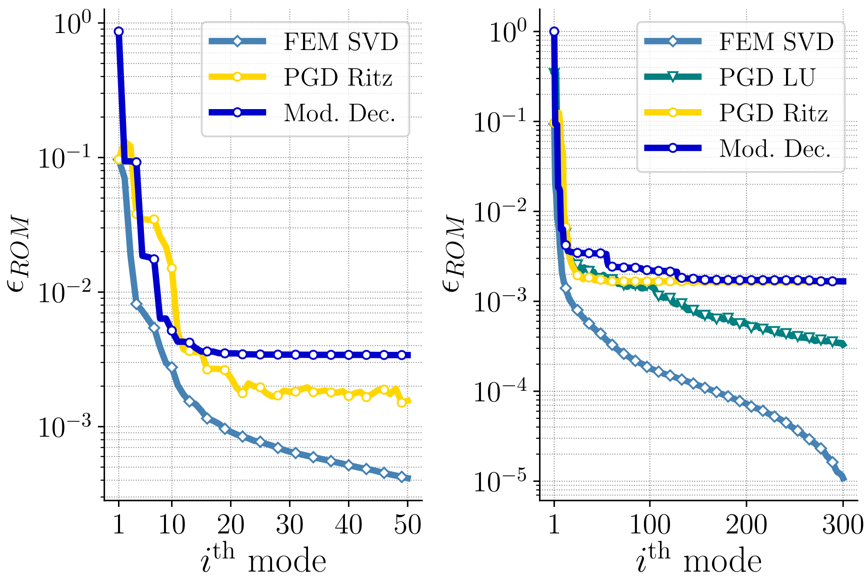


Figure 3.5 (Left) Error between the reference solutions and the SVD, PGD or Modal Decomposition approximations for 244,926 spatial DOF with 50 modes. (Right) Error between the reference solutions and the SVD, PGD or Modal Decomposition approximations for 36,774 spatial DOF with 300 modes. ($y$-axis has log scale)

– RAM: 38 GB;

– OS: Arch Linux.

The code was written using Python 3.9.17 with NumPy 1.25.0 [96] and SciPy 1.10.1 [97] built from sources and linked against BLAS/LAPACK and SuiteSparse [98].

### 3.4.3 Numerical results

**Aitken acceleration.** The relaxation technique significantly reduces the number of fixed-point iterations (see Figure 3.3). For 20 modes, Aitken acceleration saves five iterations per enrichment, on average, and a total of over 100 iterations for the full computation. Moreover, it is worth highlighting that without Aitken acceleration, the fixed-point procedure sometimes terminates without reaching convergence. This is the case for example for modes 2, 4, and 6, as shown in Figure 3.3. Indeed, the maximum number of iterations in this example is set to 35 iterations, so that if convergence is not reached within the 35 iterations, the fixed-point procedure is aborted and the last computed mode is retained. Thus, not only Aitken acceleration increases the computational efficiency, but also allows one to reach the convergence criterion that may not be satisfied otherwise. Eventually, slight discrepancies in the temporal modes may be noticeable between the results obtained with and without

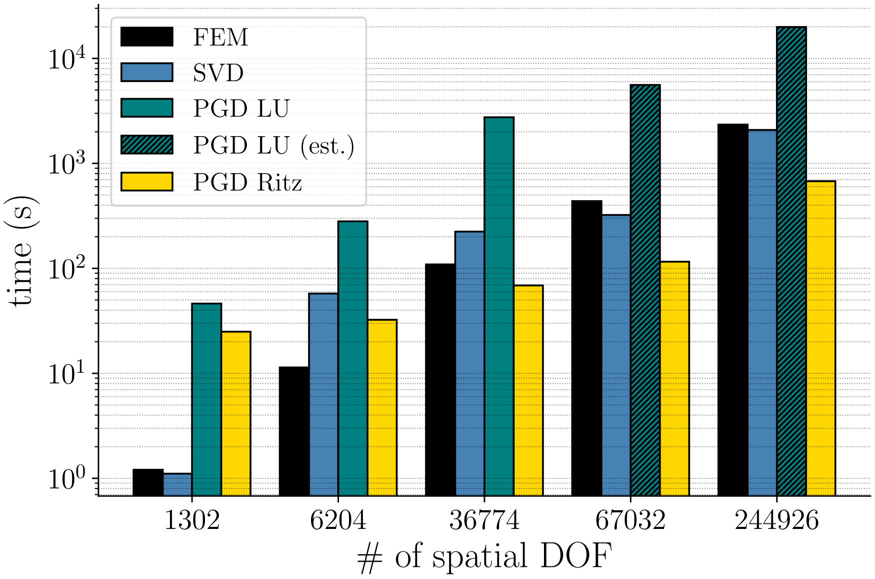


Figure 3.6 Real execution time for the full-order model (FEM) and the reduced-order models (SVD, PGD LU, PGD Ritz) with respect to different spatial discretizations ($y$-axis has log scale).

acceleration (see Figure 3.4). On the other hand, there is no significant difference on the spatial modes, as illustrated in Figure 3.8, with Aitken acceleration when using either one of the two PGD approaches.

**ROM accuracy.** Figure 3.5 shows the errors of the reduced-order models with respect to the FEM solutions for $2n = 244{,}926$ spatial degrees of freedom. We observe that the errors significantly decrease for both the PGD LU and PGD Ritz approaches during the 20 first modes. In fact, the accuracy of the PGD Ritz solution is similar to that of the PGD LU solutions. Moreover, we observe that the convergence of the two PGD approximations is comparable to that of the SVD, at least for the 20 first modes, before reaching a plateau.

**Execution time and scalability.** Figures 3.6 and 3.7 show respectively the total and detailed real execution times of the different methods. We remark that the PGD solver is not competitive when the number of degrees of freedom remains low. We also observe that, except in the case with 1,302 spatial degrees of freedom, the PGD Ritz outperforms any other method. On the one hand, the SVD, as an *a posteriori* method, requires a full-order snapshot to build a reduced-order model. Moreover, the extraction of the principal components from the data takes as much time as the actual full-order computation. On the other hand, the Ritz version of the PGD solver as an *a priori* method does not require any prior knowledge

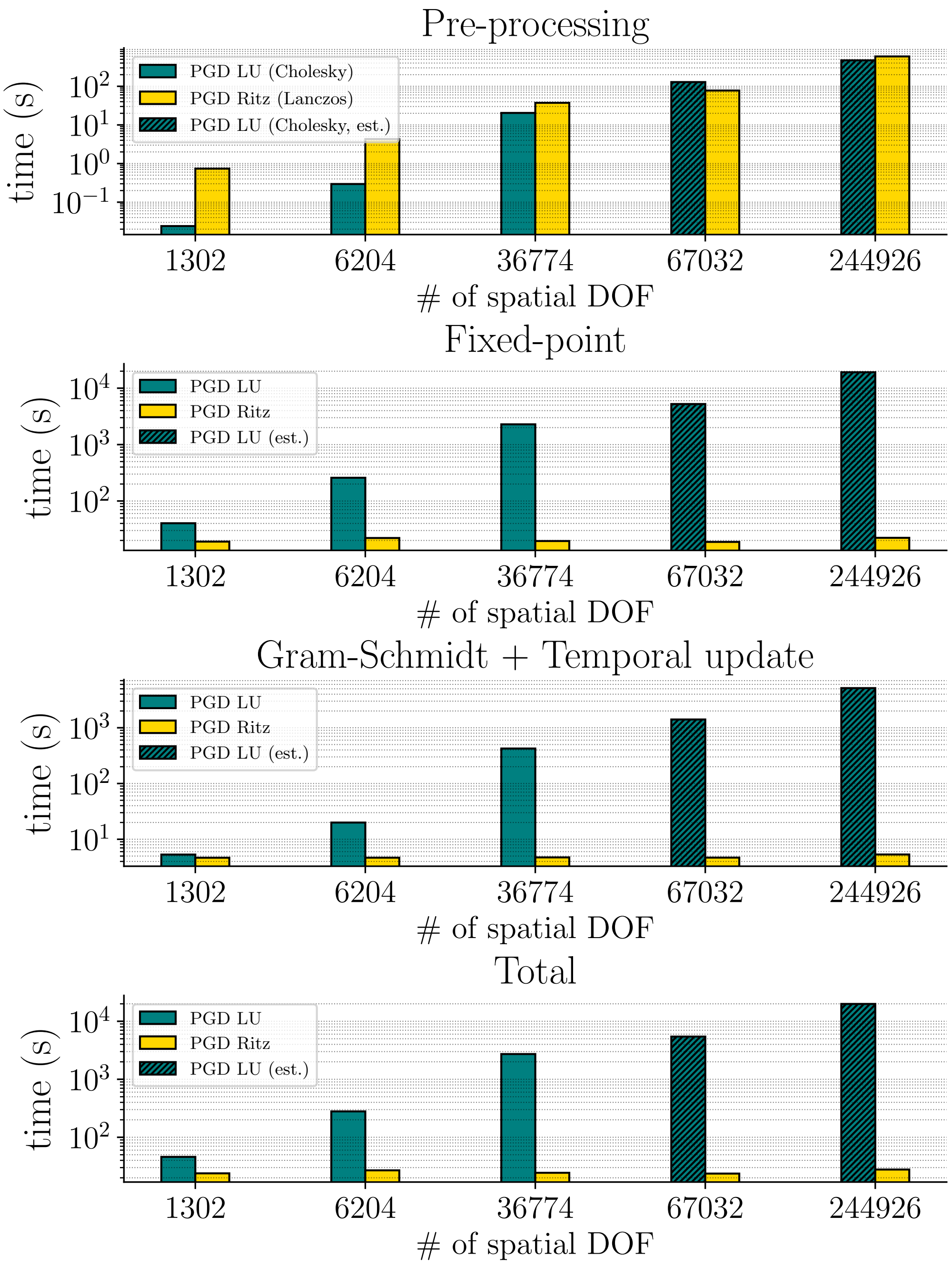


Figure 3.7 Detail on the real execution time for the full-order model (FEM) and the reduced-order models (PGD LU and PGD Ritz) with respect to different space discretizations ($y$-axis has log scale).

of the full-order solution and reaches an error comparable to that of the SVD for the first 20 modes. More precisely, the PGD Ritz does not reach an error as low as that of the SVD. However, the difference in error is small enough in view of the speedup to justify the use of the PGD Ritz over the SVD (see Table 3.1). Conversely, the use of the PGD LU in this context cannot really be justified over the SVD.

Regarding the detailed execution times, it seems that the pre-processing phase has comparable computational efficiency. In other words, the computation of a Cholesky factorization for $M$ is as costly as computing Ritz pairs. Nevertheless, carrying out the PGD computation in the subspace provided by the Ritz vectors drastically increases the performance of each of the subsequent phases, namely the fixed-point, Gram-Schmidt, and the temporal update procedures.

### 3.4.4 Further discussion

The PGD Ritz solver is overall much more efficient than the other approaches and offers a remarkably good compromise in terms of error decay. Moreover, this novel approach displays good scalability with respect to the number of spatial degrees of freedom, with a reasonable error for a relatively small number of modes, which is highly suitable in model-order reduction. The PGD Ritz solver could be interpreted as a hybrid approach between classic PGD solvers and Modal Decomposition methods. In that respect, the relevance of the PGD Ritz over classic PGD solvers is unequivocal in a space-time separated context. Yet, its advantage over Modal Decomposition must be discussed, as well as its potential to perform well if separation with additional parameters (material, geometric, etc.) had to be accounted for.

Around the 20$^{\text{th}}$ mode, we observe on Figure 3.5 that the error decay slows down or even

Table 3.1 Time efficiency of the reduced-order models (SVD, PGD LU, PGD Ritz) with respect to different spatial discretizations and PGD Ritz speedup compared to other methods.

| **# DOF in space** | $\Delta T_1$ **(s)**<br>**(FEM & SVD)** | $\Delta T_2$ **(s)**<br>**(PGD LU)** | $\Delta T_3$ **(s)**<br>**(PGD Ritz)** | **Gain** $\frac{\Delta T_1}{\Delta T_3}$ | **Gain** $\frac{\Delta T_2}{\Delta T_3}$ |
|---|---|---|---|---|---|
| 1,302 | 2.32 | 46.23 | 24.95 | 0.09 | 1.85 |
| 6,204 | 68.99 | 280.69 | 32.38 | 2.13 | 8.67 |
| 36,774 | 333.26 | 2,750.84 | 68.77 | 4.85 | 40.00 |
| 67,032 | 760.15 | n.a. | 115.79 | 6.57 | n.a. |
| 244,926 | 4,428.65 | n.a. | 676.87 | 6.54 | n.a. |

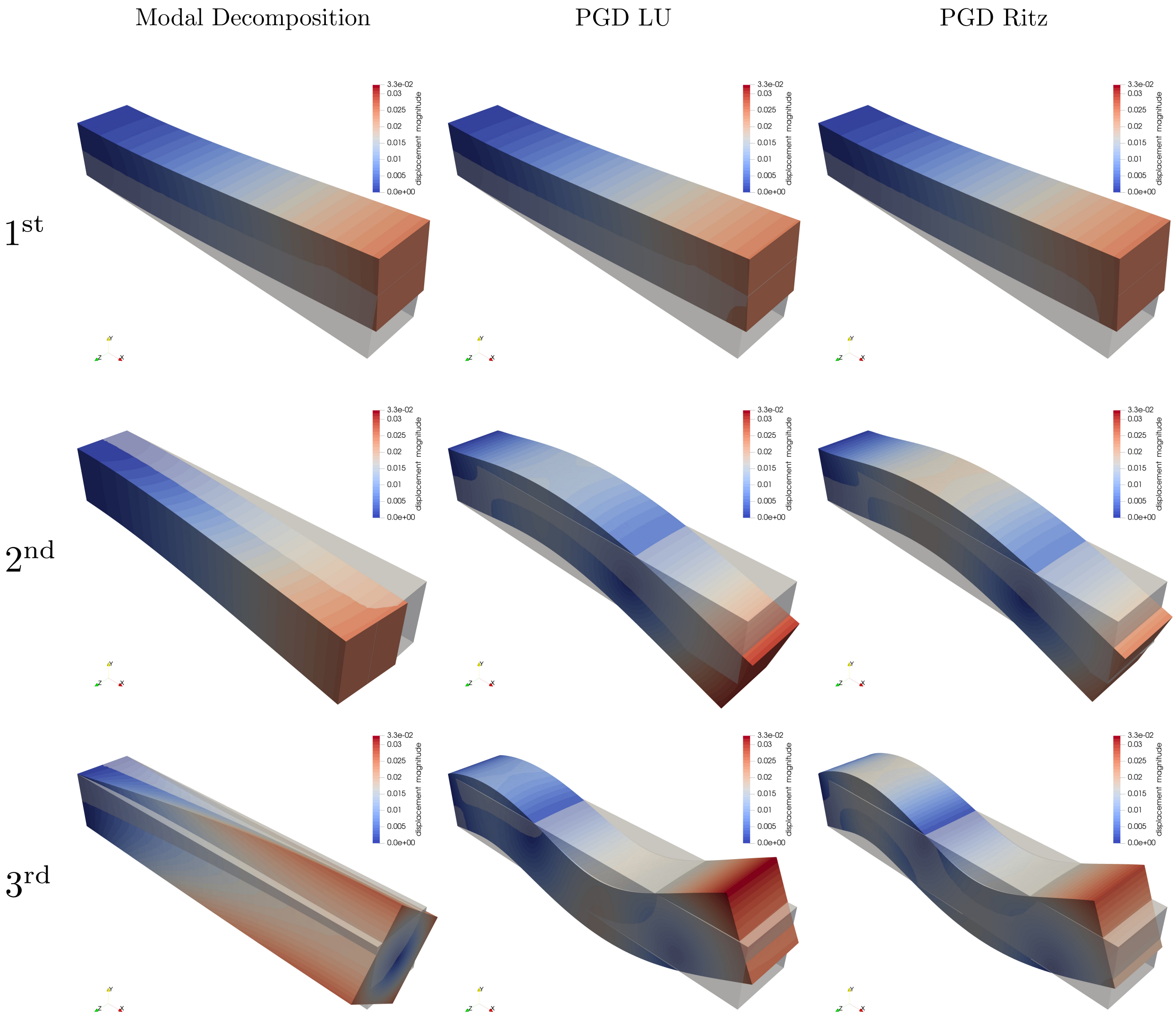


Figure 3.8 Visualization of the first three spatial modes (normalized) for the Modal Decomposition, PGD LU and PGD Ritz on the first, second and third columns, respectively and undeformed configuration in low opacity.

stagnates for the PGD Ritz. Since computations are carried out in the subspace spanned by the Ritz vectors, it is intuitively understandable that the quality of the PGD approximation is bounded by the information contained in the Ritz vectors. Indeed, Figure 3.5 illustrates this idea: the error in the solution obtained by the PGD Ritz after the first 20 enrichments matches the error of the response computed by Modal Decomposition (MD) with $r = 300$ modes (number of Ritz vectors). On the one hand, Ritz vectors are describing the natural response of the system. Thus, not all the Ritz vectors will be relevant to describe the structural response under external loads. Mode participation factors or methods such as sensitivity analysis or mode shape analysis may provide insights to select a set of vectors that capture a given dynamic behavior. However, these approaches can be tedious as they may require user intervention to interpret the results, which makes the process subjective and less repeatable. On the other hand, the PGD solver inherently accounts for external loads to compute relevant modes that describe the structural response accurately. In the PGD Ritz framework, it translates to find linear combinations of the Ritz vectors that satisfy the PGD spatial formulation (3.22) that derives from the Galerkin finite element formulation. This is well illustrated by Figure 3.8: the first three modes for Modal Decomposition are the dominant deformation modes for the beam geometry, respectively vertical bending, lateral bending, and torsion. However, for the given external load, lateral bending and torsion are not relevant. We can thus see that, like modal decomposition, the PGD solvers compute a first mode corresponding to vertical bending, but the subsequent modes also contribute to the description of vertical bending, which is effectively the dominant mode to describe the structure's response to the prescribed load.

Figure 3.5 also shows that while the error in the PGD Ritz solution reaches a plateau, that in the PGD LU solution eventually keeps decreasing when the number of modes is increased. Thus, if error stagnation is detected while the accuracy remains above a given tolerance, two strategies can be considered:

- Restarting the Arnoldi algorithm to find subsequent Ritz vectors (i.e. increase $r$), so as to enrich the research space for new PGD modes;
- Switch back to the PGD LU algorithm.

The methodology can be straightforwardly extended to viscoelastic systems modeled with Rayleigh damping, allowing for the construction of a parametric reduced-order model with respect to the Rayleigh damping coefficients. Indeed, the damping term does not change the matrix pattern of the system (3.22) to be solved for the spatial mode. In [67], the parametric eigenproblem $K_\mu \boldsymbol{u}(\boldsymbol{\mu}) = \lambda(\boldsymbol{\mu}) M_\mu \boldsymbol{u}(\boldsymbol{\mu})$ is solved for applications in structural dynamics, where the stiffness $K_\mu$ and mass $M_\mu$ operators depend on material or geometric parameters $\boldsymbol{\mu}$. The authors introduce an original method to solve this parametric eigenproblem within the PGD framework, so as to find approximations of the eigen-pairs $(\lambda(\boldsymbol{\mu}), \boldsymbol{u}(\boldsymbol{\mu}))$ in a parameter-separated format. Their approach may be considered to provide a parametrized subspace, onto which the spatial problem (3.22) can be projected to recover a diagonal structure as in (3.31). The PGD Ritz would optimize the selection of the eigenvectors that contribute to the structure response. Therefore, the PGD Ritz could present a proficient tool to compute the dynamic response of structures subjected to time-dependent loads, even in a parametrized setting.

Furthermore, the possibility to choose a symplectic time integrator in combination with the preservation of symplecticity of the spatial modes offers an appropriate foundation for a potential extension of this work. It may allow for the development of a reduction technique suited to the treatment of elastodynamics problems that involve large rotations and small strains as presented in [39]. Finally, the proposed approach allows one to consider the construction of a PGD Ritz aimed at minimizing an error with respect to a Quantity of Interest (QoI) [99]. The PGD subproblems would be modified so that combinations of the Ritz vectors are now sought for as to minimize a residual over a QoI.

## 3.5 Conclusion

The PGD solver developed here combines good accuracy and efficiency, even with an increased number of degrees of freedom. The calculation of the PGD modes in the subspace spanned by the Ritz vectors proves to be proficient, as it substantially accelerates the computation without introducing a significant approximation error. Aitken acceleration and the orthogonalization procedures are not as important for computational efficiency, but guarantee convergence and stability properties that are essential to the solver. In addition, the solver, which is based on the Hamiltonian formalism, builds reduced models for both the generalized coordinates and conjugate momenta. It has been shown that it allows the construction of a symplectic reduced basis, thus respecting the structure of canonical Hamiltonian mechanics. This is an interesting feature, as it opens up a variety of avenues related to this fundamental structure in dynamics. The numerical results also show great promise regarding the viability of this approach for solving linear elastodynamics problems on three-dimensional structures.

While the computational efficiency had been significantly increased, the accuracy of the ROM was then bounded to the subspace defined by the Ritz vectors. Consequently, the next development addressed the spatial solver in more detail, introducing a two-step prediction-correction approach to enhance the precision of the spatial modes. The parametrization of the ROM with respect to additional damping parameters will also be introduced in the next chapter.

# CHAPTER 4 EXTENSION OF THE PROPER GENERALIZED DECOMPOSITION FOR SURROGATE MODELING WITH APPLICATION TO THE IDENTIFICATION OF RAYLEIGH DAMPING PARAMETERS

In this chapter, the PGD reduced model is parametrized with respect to the Rayleigh damping coefficients $\alpha$, $\beta \in \mathbb{R}_+$. Their estimation is an active research topic in both the experimental and numerical communities [71, 72], as control of damping phenomena is a critical issue in the construction, automotive, or aerospace industries. In contrast to Modal Decomposition or Krylov-based ROM, the proposed reduced model directly incorporates the dependence on Rayleigh damping parameters in the modes. Each enrichment term includes four modes: one spatial mode, one temporal mode, and one mode for each of the damping parameters $\alpha$ and $\beta$. The treatment of the problem in space draws upon our previous work. In the offline phase, computations are initially performed in the subspace spanned by the Ritz vectors of the system. This approach offers a computationally efficient estimation of the spatial modes but limits the accuracy of the PGD approximation to the information contained in the selected Ritz vectors [100]. To enhance accuracy, the estimation is subsequently refined using a Minimal Residual iterative solver. This two-step, prediction-correction process reduces the computational cost of a full-order solution while improving the accuracy of the reduced model. Subsequently, the PGD can be used as a surrogate to perform the optimization of the damping parameters with respect to a given snapshot. A Particle Swarm Optimization (PSO) algorithm [101, 102] drives the optimization process and queries the PGD surrogate to estimate the damped response of the structure. The numerical experiment is carried out with a snapshot generated by FEM, whose damping is modeled according to the Rayleigh hypothesis. It also assumes the knowledge of the time-dependent external load. The proof of concept consists in testing the capability of the parametric PGD to: 1) build a ROM that includes damping modes; 2) employ the surrogate to identify the coefficients that were used to generate the snapshot (inverse problem). The authors in [8] pointed out that the offline phase is somewhat suboptimal when the goal is to use the ROM for optimization. Indeed, the offline phase involves an exploration of the parametric space, but it is not known a priori whether the explored regions will subsequently be exploited throughout the online phase (optimization); or worse, the optimization process could lead to assess the response of the system in some regions where the offline phase did not gather enough knowledge of. This remark especially applies to methods like the POD, where the exploration is contingent to costly, full-order solves [34]. It will be shown that the proposed method does not suffer from this issue. The first reason is that the need for full-order solutions is irrelevant here as the PGD is used as an *a priori* ROM. The second reason, of paramount importance, is that the complexity of the offline phase weakly depends on the discretization of the damping parameter spaces.

The chapter is organized as follows: in Section 4.1, we describe the model problem in its parametric form and highlight the computational burden incurred by the use of conventional FEM solvers. The PGD approach is described in Section 4.2, with a particular focus on both the complexity of the algorithm and the solution of the problem in space. In Section 4.3, we describe the combination of the PSO method with the PGD as a surrogate. Some numerical experiments are presented in Section 4.4 to illustrate the performance of the proposed approach. We finally provide some concluding remarks in Section 4.5.

## 4.1 Model problem

### 4.1.1 Strong formulation

The model problem we shall consider is that of elastodynamics in three dimensions under the assumption of infinitesimal deformation and linear viscoelastic damping. Let $\Omega$ be an open bounded domain of $\mathbb{R}^3$, with Lipschitz boundary $\partial\Omega$, and let $\mathcal{I} = (0, T)$ denote the time interval. The boundary $\partial\Omega$ is supposed to be decomposed into two portions, $\partial\Omega_D$ and $\partial\Omega_N$, such that $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. The displacement field $u : \bar{\Omega} \times \bar{\mathcal{I}} \to \mathbb{R}^3$ satisfies the following partial differential equation:

$$\rho \frac{\partial^2 u}{\partial t^2} - \nabla \cdot \sigma(u) = f, \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{4.1}$$

where, in the case of infinitesimal deformation, the stress tensor $\sigma(u)$ and strain tensor $\varepsilon(u)$ are given by:

$$\sigma(u) = \mathbb{E} : \varepsilon(u) + \mathbb{D} : \dot{\varepsilon}(u), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{4.2}$$

$$\varepsilon(u) = \frac{1}{2}\Big(\nabla u + \big(\nabla u\big)^T\Big), \qquad \forall (x,t) \in \Omega \times \mathcal{I}, \tag{4.3}$$

and is subjected to the initial conditions:

$$u(x,0) = u_0(x), \qquad \forall x \in \Omega, \tag{4.4}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \qquad \forall x \in \Omega, \tag{4.5}$$

as well as to the boundary conditions:

$$u(x,t) = 0, \qquad \forall (x,t) \in \partial\Omega_D \times \mathcal{I}, \tag{4.6}$$

$$\sigma(u) \cdot n = g_N(x,t), \qquad \forall (x,t) \in \partial\Omega_N \times \mathcal{I}. \tag{4.7}$$

The functions $f : \Omega \times \mathcal{I} \to \mathbb{R}^3$, $u_0 : \Omega \to \mathbb{R}^3$, $v_0 : \Omega \to \mathbb{R}^3$, and $g_N : \partial\Omega_N \times \mathcal{I} \to \mathbb{R}^3$ are supposed to be sufficiently regular to yield a well-posed problem. The tensors $\mathbb{E}$ and $\mathbb{D}$ describe the elastic and viscous local properties, respectively, of the medium occupied in $\Omega$. The latter is assumed to be isotropic, with density $\rho$ and Lamé coefficients $\lambda$ and $\mu$ (we note that the material parameters could possibly vary in space). The constitutive equation (4.2) thus reduces to:

$$\sigma(u) = 2\mu\varepsilon(u) + \lambda \mathrm{tr}\,(\varepsilon(u))\, I_3 + \mathbb{D} : \dot{\varepsilon}(u),$$

where $I_3 \in \mathbb{R}^{3\times3}$ is the identity matrix. In the following, we will denote the first and second time derivatives by $\dot{u} = \partial u/\partial t$ and $\ddot{u} = \partial^2 u/\partial t^2$.

### 4.1.2 Semi-weak formulation

We consider here the semi-weak formulation with respect to the spatial variable in order to construct the discrete problem in space using the Finite Element method (FEM). Multiplying (4.1) by an arbitrary smooth function $u^* = u^*(x)$ and integrating over the whole domain $\Omega$, one obtains:

$$\int_\Omega \rho\ddot{u} \cdot u^* - (\nabla \cdot \sigma(u)) \cdot u^* \; dx = \int_\Omega f \cdot u^* \; dx, \qquad \forall t \in \mathcal{I}. \tag{4.8}$$

Following the same development as in [100], the semi-discrete formulation of the problem then reads:

Find $u$ such that for all $t \in \mathcal{I}$, $u(\cdot,t) \in V$, and

$$\begin{aligned} &\int_\Omega \rho\ddot{u} \cdot u^* + \varepsilon(u) : \mathbb{E} : \varepsilon(u^*) + \dot{\varepsilon}(u) : \mathbb{D} : \varepsilon(u^*) \; dx \\ &\qquad = \int_\Omega f \cdot u^* \; dx + \int_{\partial\Omega_N} g_N \cdot u^* \; dx, \qquad \forall u^* \in V, \quad \forall t \in \mathcal{I}, \end{aligned} \tag{4.9}$$

and

$$u(x,0) = u_0(x), \qquad \forall x \in \Omega,$$

$$\dot{u}(x,0) = v_0(x), \qquad \forall x \in \Omega,$$

where $V$ is the vector space of vector-valued functions defined on $\Omega$:

$$V = \left\{ v \in \left[ H^1(\Omega) \right]^3 : \ v = 0 \text{ on } \partial\Omega_D \right\}.$$

### 4.1.3 Spatial discretization

We partition the domain into $N_e$ tetrahedral elements $K_e$ such that $\overline{\Omega} = \cup_{e=1}^{N_e} K_e$ and $\mathrm{Int}(K_i) \cap \mathrm{Int}(K_j) = \varnothing$, $\forall i, j = 1, \dots, N_e$, $i \neq j$. We then associate with the mesh the finite-dimensional Finite Element space $W^h$, dim $W^h = s$, of scalar-valued continuous and piecewise polynomial functions defined on $\Omega$, that is:

$$W^h = \left\{ v_h : \Omega \to \mathbb{R} : \ v_h|_{K_e} \in \mathbb{P}_k(K_e), \ e = 1, \dots, N_e \right\},$$

where $\mathbb{P}_k(K_e)$ denotes the space of polynomial functions of degree $k$ on $K_e$. Let $\phi_i$, $i = 1, \dots, s$, denote the basis functions of $W^h$, i.e. $W^h = \mathrm{span}\{\phi_i\}$. We then introduce the finite element subspace $V^h$ of $V$ such as:

$$V^h = \left[ W^h \right]^3 \cap V,$$

and search for finite element solutions $u_h$ satisfying $u_h(\cdot, t) \in V^h$, $\forall t \in \bar{\mathcal{I}}$, in the form:

$$u_h(x,t) = \sum_{j=1}^{s} \phi_j(x) q_j(t),$$

where the vectors of degrees of freedom, $q_j \in \mathbb{R}^3$, depend on time. We introduce the set of $n = 3s$ vector-valued basis functions as:

$$\chi_{3i-2}(x) = \begin{bmatrix} \phi_i(x) \\ 0 \\ 0 \end{bmatrix}, \quad \chi_{3i-1}(x) = \begin{bmatrix} 0 \\ \phi_i(x) \\ 0 \end{bmatrix}, \quad \chi_{3i}(x) = \begin{bmatrix} 0 \\ 0 \\ \phi_i(x) \end{bmatrix}, \qquad i = 1, \dots, s.$$

Using the Galerkin method, the Finite Element problem thus reads:

Find $u_h$ such that $u_h(\cdot, t) \in V^h$, $\forall t \in \bar{\mathcal{I}}$, and

$$\int_\Omega \rho \chi_i(x) \cdot \ddot{u}_h(x,t) + \varepsilon(\chi_i)(x) : \mathbb{E} : \varepsilon(u_h)(x,t) + \varepsilon(\chi_i)(x) : \mathbb{D} : \dot{\varepsilon}(u_h)(x,t) \, dx$$
$$= \int_\Omega \chi_i(x) \cdot f(x,t) \, dx + \int_{\partial\Omega_N} \chi_i(x) \cdot g_N(x,t) \, ds, \quad \forall i = 1, \dots, n, \quad \forall t \in \mathcal{I},$$

satisfying the initial conditions

$$u_h(x,0) = u_{0,h}(x), \quad \forall x \in \Omega,$$
$$\dot{u}_h(x,0) = v_{0,h}(x), \quad \forall x \in \Omega,$$

where $u_{0,h}$ and $v_{0,h}$ are interpolants or projections of $u_0$ and $v_0$ in the space $V^h$. The above problem can be conveniently recast in compact form as:

$$M\ddot{\boldsymbol{q}}(t) + C\dot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{4.10}$$

$$\boldsymbol{q}(0) = \boldsymbol{u}_0, \tag{4.11}$$

$$\dot{\boldsymbol{q}}(0) = \boldsymbol{v}_0, \tag{4.12}$$

where $M$, $C$, and $K$ are the global mass, damping, and stiffness matrices, respectively:

$$\begin{aligned} M_{ij} &= \int_\Omega \rho\chi_i \cdot \chi_j \; dx, \\ K_{ij} &= \int_\Omega \varepsilon(\chi_i) : \mathbb{E} : \varepsilon(\chi_j) \; dx \;, \qquad \forall i, j = 1, \ldots, n, \\ C_{ij} &= \int_\Omega \varepsilon(\chi_i) : \mathbb{D} : \varepsilon(\chi_j) \; dx, \end{aligned}$$

$\boldsymbol{f}(t)$ is the loading vector at time $t$ whose components are given by:

$$f_i(t) = \int_\Omega \chi_i(x) \cdot f(x,t)\, dx + \int_{\partial\Omega_N} \chi_i(x) \cdot g_N(x,t) \; ds, \qquad \forall i = 1, \ldots, n,$$

$\boldsymbol{q}(t)$ is the global vector of degrees of freedom:

$$\boldsymbol{q}(t) = \begin{bmatrix} q_1(t) & \ldots & q_s(t) \end{bmatrix}^T,$$

where $q_i \in \mathbb{R}^3$, $i = 1, \ldots, s$, and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial vectors:

$$\begin{aligned} \boldsymbol{u}_0 &= \begin{bmatrix} u_{0,1} & \ldots & u_{0,s} \end{bmatrix}^T, \\ \boldsymbol{v}_0 &= \begin{bmatrix} v_{0,1} & \ldots & v_{0,s} \end{bmatrix}^T. \end{aligned}$$

Note that $u_{0,i} \in \mathbb{R}^3$ and $v_{0,i} \in \mathbb{R}^3$, $i = 1, \ldots, s$, are vectors whose components are the initial displacements and velocities in the three spatial directions. Moreover, the above definition of $C$ is not practical and will be replaced by the model described in the next section.

### 4.1.4 Damping modeling and parameter discretization

We assume here that damping is “proportional” to $K$ and $M$, that is, $C = \alpha K + \beta M$, where $\alpha$, $\beta \in \mathbb{R}_+$ are the so-called Rayleigh damping coefficients. Furthermore, we shall rewrite the damping matrix as follows:

$$C(\alpha, \beta) = \alpha\bar{\alpha}K + \beta\bar{\beta}M, \tag{4.13}$$

where the constant parameters $\bar{\alpha}$, $\bar{\beta} \in \mathbb{R}_+$ are interpreted as reference parameters while $\alpha$ and $\beta$ encapsulate the variability in the parameters. Let $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ be spaces for parameters $\alpha$, $\beta \in \mathbb{R}_+$, respectively, such that:

$$\mathcal{D}_\alpha = [\alpha_{\min}, \alpha_{\max}],$$
$$\mathcal{D}_\beta = [\beta_{\min}, \beta_{\max}].$$

Let $\mathcal{D}$ be the global parameter space defined as the tensor product of $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$, i.e. $\mathcal{D} = \mathcal{D}_\alpha \times \mathcal{D}_\beta$. Our goal is to compute the mechanical response of a system over $\Omega \times \mathcal{I} \times \mathcal{D}$. Finite element solutions are sought for in the form:

$$u_h(x, t, \alpha, \beta) = \sum_{j=1}^{s} \phi_j(x) q_j(t, \alpha, \beta).$$

In other words, we aim at modeling the mechanical behavior for multiple pairs of values $(\alpha, \beta) \in \mathcal{D}$, $(\alpha, \beta)$ being chosen so that the computed approximations are in the following solution manifold:

$$\mathcal{M} = \left\{ v_h : \mathcal{D} \to W^h \otimes L^2(\mathcal{I});\ \int_{\mathcal{D}} |||v_h|||^2 d\mu < \infty \right\},$$

where:

$$|||u||| = \sqrt{\int_{\mathcal{I}} \int_{\Omega} \frac{1}{2} \rho \dot{u} \cdot \dot{u} + \frac{1}{2} \varepsilon(u) : \mathbb{E} : \varepsilon(u)\ dxdt}.$$

Later on, the parameter spaces $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ will be partitioned into $(n_\alpha - 1)$ and $(n_\beta - 1)$ subintervals, respectively, that is, $\mathcal{D}_\alpha = \cup_{i=2}^{n_\alpha} [\alpha_{i-1}, \alpha_i]$ and $\mathcal{D}_\beta = \cup_{i=2}^{n_\beta} [\beta_{i-1}, \beta_i]$. Here, $n_\alpha$ and $n_\beta$ denote the numbers of grid points $\alpha_i$, $i = 1, \ldots, n_\alpha$, and $\beta_i$, $i = 1, \ldots, n_\beta$, chosen in $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$, respectively. We also define $\|\cdot\|_{\mathcal{M}}$ such that:

$$\forall u \in \mathcal{M}, \quad \|u\|_{\mathcal{M}} = \int_{\mathcal{D}} |||u|||^2 d\mu.$$

In many applications, it is useful to know the mechanical response for many pairs of parameters $(\alpha, \beta)$ in the global parameter space $\mathcal{D}$. However, this entails a substantial computational burden when using conventional methods, as shown in the next section.

### 4.1.5 Discretization in time

The time domain $\mathcal{I}$ is divided into $n_t$ subintervals $\mathcal{I}^i = [t^{i-1}, t^i]$, $i = 1, \ldots, n_t$, of size $h_i = t^i - t^{i-1}$. For the sake of simplicity, we assume here that all intervals are of same length

$h_t$, i.e. $h_i = h_t$ for all $i = 1, \ldots, n_t$. Introducing the vector $\boldsymbol{p}(t) = M\dot{\boldsymbol{q}}(t)$, Eq. (4.10) can be recast into the system of coupled equations:

$$\dot{\boldsymbol{p}}(t) + C\dot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{4.14}$$

$$\dot{\boldsymbol{q}}(t) - M^{-1}\boldsymbol{p}(t) = 0, \qquad \forall t \in \mathcal{I}. \tag{4.15}$$

The two equations (4.14) and (4.15) are discretized using the Crank-Nicolson scheme (also referred to as the implicit trapezoidal rule) to obtain, for $i = 1, \ldots, n_t$:

$$\begin{bmatrix} h_tK + 2C & 2I_n \\ 2I_n & -h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^i \\ \boldsymbol{p}^i \end{bmatrix} = \begin{bmatrix} -h_tK + 2C & 2I_n \\ 2I_n & h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{i-1} \\ \boldsymbol{p}^{i-1} \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{f}^i + \boldsymbol{f}^{i-1} \\ 0 \end{bmatrix}, \tag{4.16}$$

with:

$$\begin{aligned} \boldsymbol{q}^0 &= \boldsymbol{u}_0, \\ \boldsymbol{p}^0 &= M\boldsymbol{v}_0. \end{aligned}$$

Eq. (4.16) can be rewritten as:

$$\begin{bmatrix} h_tK + 2C & 2I_n \\ 2I_n & -h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^i \\ \boldsymbol{p}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_q^i \\ \boldsymbol{b}_p^i \end{bmatrix},$$

with

$$\begin{aligned} \boldsymbol{b}_q^i &= [-h_tK + 2C]\,\boldsymbol{q}^{i-1} + 2\boldsymbol{p}^{i-1} + h_t\left(\boldsymbol{f}^i + \boldsymbol{f}^{i-1}\right), \\ \boldsymbol{b}_p^i &= 2\boldsymbol{q}^{i-1} + h_tM^{-1}\boldsymbol{p}^{i-1}. \end{aligned}$$

Developing above system yields the two equations:

$$\begin{aligned} [h_tK + 2C]\,\boldsymbol{q}^i + 2\boldsymbol{p}^i &= \boldsymbol{b}_q^i, \\ 2\boldsymbol{q}^i - h_tM^{-1}\boldsymbol{p}^i &= \boldsymbol{b}_p^i. \end{aligned}$$

Rewriting the second equation as:

$$\boldsymbol{p}^i = \frac{2}{h_t}M\boldsymbol{q}^i - \frac{1}{h_t}M\boldsymbol{b}_p^i,$$

and substituting $\boldsymbol{p}^i$ for this new expression in the first equation leads to:

$$\left[h_tK + 2C + \frac{4}{h_t}M\right]\boldsymbol{q}^i = \boldsymbol{b}_q^i + \frac{2}{h_t}M\boldsymbol{b}_p^i.$$

Finally, applying the Rayleigh hypothesis for $C$ (4.13), one has to solve at each time-step the coupled equations:

$$\left[(h_t + 2\alpha\bar{\alpha})K + \left(\frac{4}{h_t} + 2\beta\bar{\beta}\right) M\right] \boldsymbol{q}^i = \boldsymbol{b}_q^i + \frac{2}{h_t} M \boldsymbol{b}_p^i, \tag{4.17}$$

$$\boldsymbol{p}^i = \frac{2}{h_t} M \boldsymbol{q}^i - \frac{1}{h_t} M \boldsymbol{b}_p^i. \tag{4.18}$$

Since $K$ and $M$ are both symmetric and positive-definite, so is the left-hand side matrix in (4.17). Therefore, the equation can be solved at the cost of one Cholesky factorization of the matrix $A = \left[(h_t + 2\alpha\bar{\alpha})K + \left(4h_t^{-1} + 2\beta\bar{\beta}\right)M\right]$ (as long as the time-step $h_t$ remains constant), while all subsequent operations inside one time-step loop will have costs similar to those of matrix-vector multiplications.

The corresponding algorithm is shown in Algorithm 4. Its complexity depends on the number of non-zero elements in the matrices that are dealt with and on their sparsity patterns (note that $K$ and $M$ feature the same number of non-zero elements and sparsity patterns, properties that are directly inherited by $A$). As an approximation, if we consider that all matrices are dense, of size $n$, the cost of Algorithm 4 is thus proportional to that of a Cholesky factorization, i.e. $\mathcal{O}\left(n^3\right)$, and $n_t$ matrix-vector multiplications, each having a cost of $\mathcal{O}\left(n^2\right)$. The resulting complexity of the algorithm is therefore of the order of $\mathcal{O}\left(n^3 + n_t n^2\right)$. We also note that the Cholesky factorization is the most expensive operation, but that the integrating scheme could also be costly if $n_t$ becomes large.

Accounting for parameters, the evaluation of the mechanical response over the discrete global parameter space $\mathcal{D}$ by means of Algorithm 4 has complexity of order $\mathcal{O}\left(n_\alpha n_\beta \left(n^3 + n_t n^2\right)\right)$. It shows that the complexity of such a task rapidly grows with respect to the number of parameters. We describe below a reduced-order technique that efficiently tackles the evaluation of $u_h$ when dealing with many parameters. Furthermore, we will denote the complexity of solving a linear system of $n$ algebraic equations in $n$ unknowns as $\text{lin}(n)$. This notation is used to generalize the complexity, whether a direct or iterative solver is employed, and regardless of the sparsity pattern of the operators.

## 4.2 Parametric PGD reduced-order modeling

The Proper Generalized Decomposition method aims at approximating both the generalized coordinates $\boldsymbol{q}$ and their generalized momenta $\boldsymbol{p}$ in separated form. We are thus searching for

**Algorithm 4** Classic elastodynamics FEM-Crank-Nicolson solver

```
Input: assembled operators $K$, $M$ and $\boldsymbol{f}$
          initial conditions $\boldsymbol{q}^0$ and $\boldsymbol{p}^0$
Define $A = \left[(h_t + 2\alpha\bar{\alpha})K + \left(4h_t^{-1} + 2\beta\bar{\beta}\right) M\right]$
Factorize $A$: L = cholesky($A$)          see sksparse.cholmod [98] sparse Cholesky decomposition
for $i$=1, ..., $n_t$ do
    Update $\boldsymbol{b}_q^i$ and $\boldsymbol{b}_p^i$
    Solve $\boldsymbol{q}^i$ = L.solve_A($\boldsymbol{b}_q^i + 2h_t^{-1}M\boldsymbol{b}_p^i$) forward / backward substitution for Eq. (4.17)
    Compute $\boldsymbol{p}^i$                                         Eq. (4.18)
end for
Output: $Q = \left[\boldsymbol{q}^0 \;\dots\; \boldsymbol{q}^{n_t}\right]$ and $P = \left[\boldsymbol{p}^0 \;\dots\; \boldsymbol{p}^{n_t}\right]$
```

a space-time separated representation of $\boldsymbol{q}$ and $\boldsymbol{p}$ as:

$$\begin{aligned}
\boldsymbol{q}(t,\alpha,\beta) \approx \boldsymbol{q}_m(t,\alpha,\beta) &= \sum_{i=1}^{m} \boldsymbol{\varphi}_i^q \psi_i^q(t)\xi_i^q(\alpha)\zeta_i^q(\beta),\\
\boldsymbol{p}(t,\alpha,\beta) \approx \boldsymbol{p}_m(t,\alpha,\beta) &= \sum_{i=1}^{m} \boldsymbol{\varphi}_i^p \psi_i^p(t)\xi_i^p(\alpha)\zeta_i^p(\beta).
\end{aligned}$$

For the sake of clarity in the presentation, we shall drop from now on the subscript $i$ and write the decompositions of rank $m$ as:

$$\begin{aligned}
\boldsymbol{q}(t,\alpha,\beta) \approx \boldsymbol{q}_m(t,\alpha,\beta) &= \boldsymbol{q}_{m-1}(t,\alpha,\beta) + \boldsymbol{\varphi}_q \psi_q(t)\xi_q(\alpha)\zeta_q(\beta),\\
\boldsymbol{p}(t,\alpha,\beta) \approx \boldsymbol{p}_m(t,\alpha,\beta) &= \boldsymbol{p}_{m-1}(t,\alpha,\beta) + \boldsymbol{\varphi}_p \psi_p(t)\xi_p(\alpha)\zeta_p(\beta).
\end{aligned}$$

The bold notations $\boldsymbol{z}$, $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$, $\boldsymbol{\xi}$, and $\boldsymbol{\zeta}$ denote vertical concatenations such that:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}, \quad \boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\varphi}_q \\ \boldsymbol{\varphi}_p \end{bmatrix}, \quad \boldsymbol{\psi} = \begin{bmatrix} \psi_q \\ \psi_p \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} \xi_q \\ \xi_p \end{bmatrix}, \quad \boldsymbol{\zeta} = \begin{bmatrix} \zeta_q \\ \zeta_p \end{bmatrix}.$$

The following weighted residual form of (4.14)–(4.15) is considered:

$$\begin{aligned}
\int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{q}^{*T}\left[\dot{\boldsymbol{p}} + \left(\alpha\bar{\alpha}K + \beta\bar{\beta}M\right)\dot{\boldsymbol{q}} + K\boldsymbol{q}\right] dt d\mu &= \int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{q}^{*T}\boldsymbol{f}\; dt d\mu, \quad && \forall \boldsymbol{q}^* \in L^2(\mathcal{I}, V^h),\\
\int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{p}^{*T}\left[\dot{\boldsymbol{q}} - M^{-1}\boldsymbol{p}\right] dt d\mu &= 0, && \forall \boldsymbol{p}^* \in L^2(\mathcal{I}, V^h).
\end{aligned}$$

The separated representation is computed in a progressive manner by adding one quadruplet of modes $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ at each enrichment, following the so-called greedy rank-one update algorithm. In other words, we substitute $\boldsymbol{q}_m$ and $\boldsymbol{p}_m$ for $\boldsymbol{q}$ and $\boldsymbol{p}$, respectively, so that the modes $(\boldsymbol{\varphi}_q, \psi_q, \xi_q, \zeta_q)$ and $(\boldsymbol{\varphi}_p, \psi_p, \xi_p, \zeta_p)$ become the new unknowns of the problem. Subsequently, the modes are computed one by one at each enrichment, with $\boldsymbol{z}_{m-1}$ being known. This computing paradigm has the virtue of drastically reducing the complexity by splitting the computational costs into lower dimensional sub-problems.

The goal in this section is to show how the PGD sub-problems are modified in comparison to our previous work [100] with the introduction of additional modes for parameters $\alpha$ and $\beta$. This development is rather straightforward and is drawn from an example with the transient diffusion equation in [85], where the diffusivity is considered as an extra-coordinate. The authors highlight that the analytical solution of the transient diffusion equation cannot be written in a parameter-separated format. Yet, the PGD strategy is somehow efficient to tackle the diffusion problem. Likewise, modal superposition does not express the solution of the problem that is therein dealt with in a separated form [15]. We will show in Section 4.4 that the PGD was found to be efficient when applied to the problem at hand. Special attention will be devoted to the way we address the spatial problem and a new resolution strategy will be presented. The effect of the PGD technique on the complexity will be detailed as well.

### 4.2.1 Fixed-point algorithm

If the complexity reduction offered by the PGD strategy is appealing, it however leads to a non-linear formulation for the modes. At each enrichment $m$, the problem to be solved for $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ reads:

$$\begin{aligned}\int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{q}^{*T}\left[\boldsymbol{\varphi}_p\dot{\psi}_p\xi_p\zeta_p + \left(\alpha\bar{\alpha}K + \beta\bar{\beta}M\right)\boldsymbol{\varphi}_q\dot{\psi}_q\xi_q\zeta_q + K\boldsymbol{\varphi}_q\psi_q\xi_q\zeta_q\right] dtd\mu \\ = \int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{q}^{*T}\boldsymbol{r}^q_{m-1}\, dtd\mu,\ \forall \boldsymbol{q}^*, \quad (4.19) \\ \int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{p}^{*T}\left[\boldsymbol{\varphi}_q\dot{\psi}_q\xi_q\zeta_q - M^{-1}\boldsymbol{\varphi}_p\psi_p\xi_p\zeta_p\right] dtd\mu = \int_{\mathcal{D}}\int_{\mathcal{I}} \boldsymbol{p}^{*T}\boldsymbol{r}^p_{m-1}\, dtd\mu,\ \forall \boldsymbol{p}^*,\end{aligned}$$

where $\boldsymbol{r}^q_{m-1}$ and $\boldsymbol{r}^p_{m-1}$ are the $m^{\text{th}}$ residuals:

$$\boldsymbol{r}^q_{m-1} = \begin{cases}\boldsymbol{f}, & \text{if } m = 1,\\ \boldsymbol{f} - \dot{\boldsymbol{p}}_{m-1} - \left(\alpha\bar{\alpha}K + \beta\bar{\beta}M\right)\dot{\boldsymbol{q}}_{m-1} - K\boldsymbol{q}_{m-1}, & \text{if } m > 1,\end{cases}$$

$$\boldsymbol{r}^p_{m-1} = \begin{cases}\boldsymbol{0}, & \text{if } m = 1,\\ M^{-1}\boldsymbol{p}_{m-1} - \dot{\boldsymbol{q}}_{m-1}, & \text{if } m > 1.\end{cases}$$

The above problem is solved in an iterative manner using the following fixed point algorithm:

1. Solve (4.19) for $\boldsymbol{\varphi}$ with $(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ known. This step will be referred to as the spatial problem and is written in a generic form as:

$$A_{\mathcal{S}}(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})\boldsymbol{\varphi} = \boldsymbol{b}_{\mathcal{S}}(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{4.20}$$

   where the $2n \times 2n$ matrix $A_{\mathcal{S}}$ and vector $\boldsymbol{b}_{\mathcal{S}}$ of size $2n$ will be specified in Section 4.2.2.

2. Solve (4.19) for $\boldsymbol{\xi}$ with $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta})$ known. This problem consists in solving the $2n_\alpha$ algebraic equations:

$$A_\xi(\alpha, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta})\boldsymbol{\xi} = \boldsymbol{b}_\xi(\alpha, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{4.21}$$

   where the $2 \times 2$ matrix $A_\xi$ and vector $\boldsymbol{b}_\xi$ of size 2 are explicitly provided in Appendix B. In other words, there is one $2 \times 2$ linear system to solve for each value of the parameter $\alpha$.

3. Solve (4.19) for $\boldsymbol{\zeta}$ with $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \xi)$ known. This problem consists in solving the $2n_\beta$ algebraic equations:

$$A_\zeta(\beta, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi})\boldsymbol{\zeta} = \boldsymbol{b}_\zeta(\beta, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{z}_{m-1}), \tag{4.22}$$

   where the $2 \times 2$ matrix $A_\zeta$ and vector $\boldsymbol{b}_\zeta$ of size 2 are explicitly provided in Appendix B. As for $\alpha$, there is one $2 \times 2$ linear system to solve for each value of the parameter $\beta$.

4. Solve (4.19) for $\boldsymbol{\psi}$ with $(\boldsymbol{\varphi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ known. The temporal problem corresponds to the system of first-order differential equations:

$$\dot{\boldsymbol{\psi}} = f_{\mathcal{T}}(t, \boldsymbol{\psi}, \boldsymbol{\varphi}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{4.23}$$

   where the vector-valued function $f_{\mathcal{T}}$ is explicitly provided in Appendix B.

The fixed-point procedure is illustrated in Figure 4.1. It appears that the resulting complexity of the PGD algorithm is of order $\mathcal{O}\left(mk_{\max}\left(\text{lin}(n) + n_t + n_\alpha + n_\beta\right)\right)$, with $k_{\max}$ the maximum number of fixed-point iterations to reach convergence. We will show in the next section that this complexity can further be decreased by projecting the spatial problem onto a subspace spanned by Ritz vectors.

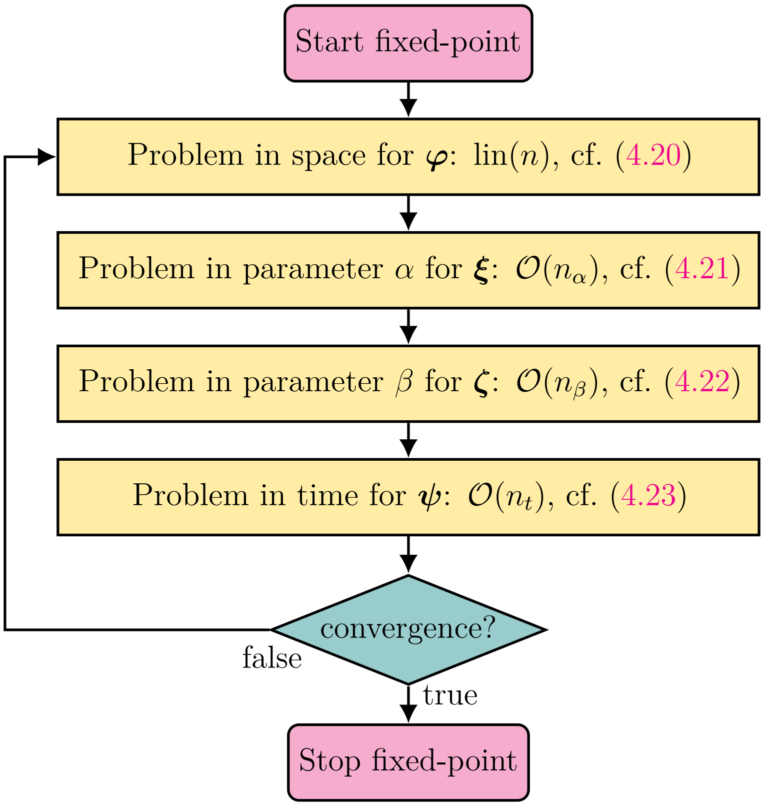

Start fixed-point
Problem in space for $\boldsymbol{\varphi}$: lin($n$), cf. (4.20)
Problem in parameter $\alpha$ for $\boldsymbol{\xi}$: $\mathcal{O}(n_\alpha)$, cf. (4.21)
Problem in parameter $\beta$ for $\boldsymbol{\zeta}$: $\mathcal{O}(n_\beta)$, cf. (4.22)
Problem in time for $\boldsymbol{\psi}$: $\mathcal{O}(n_t)$, cf. (4.23)
convergence?
false
true
Stop fixed-point

Figure 4.1 Flowchart of the conventional fixed-point algorithm with associated computational complexities.

### 4.2.2 Problem in space

#### Projection in Ritz subspace

We assume that $(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ is known and we search for the new spatial mode $\boldsymbol{\varphi}$. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q^* \psi_q \xi_q \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p^* \psi_p \xi_p \zeta_p$. Equation (4.19) leads to the linear system:

$$A_{\mathcal{S}} \boldsymbol{\varphi} = \boldsymbol{b}_{\mathcal{S}}, \tag{4.24}$$

where:

$$A_{\mathcal{S}} = \begin{bmatrix} k_{qq} K + m_{qq} M & c_{qp} I_n \\ c_{pq} I_n & m_{pp} M^{-1} \end{bmatrix},$$

$$\boldsymbol{b}_{\mathcal{S}} = \begin{bmatrix} \displaystyle\int_{\mathcal{D}} \int_{\mathcal{I}} \psi_q \xi_q \zeta_q \boldsymbol{r}_{m-1}^q \; dt d\mu \\ \displaystyle\int_{\mathcal{D}} \int_{\mathcal{I}} \psi_p \xi_p \zeta_p \boldsymbol{r}_{m-1}^p \; dt d\mu \end{bmatrix},$$

and the real coefficients are defined in Appendix B. The introduction of viscous damping, assumed to be proportional to $K$ and $M$, allows us to write (4.24) in a form similar to that in our previous work [100]. Therefore, following [100], we can project Eq. (4.24) onto the subspace of approximated eigenvectors, namely the Ritz vectors, which verify the properties (with $m \leqslant r \ll n$):

$$\left(\hat{\Lambda}, \hat{V}\right) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{n\times r}, \quad \text{such that} \quad \hat{V}^T K \hat{V} = \hat{\Lambda}, \quad \text{and} \quad \hat{V}^T M \hat{V} = I_r,$$

where the Ritz values and associated Ritz vectors are:

$$\begin{aligned} \hat{\Lambda} &= \operatorname{diag}\left(\hat{\lambda}_1, \ \ldots \ , \hat{\lambda}_r\right), \\ \hat{V} &= \left[\hat{\boldsymbol{v}}_1 \ \ldots \ \hat{\boldsymbol{v}}_r\right]. \end{aligned}$$

We now introduce the mapping:

$$R = \begin{bmatrix} \hat{V} & 0 \\ 0 & M\hat{V} \end{bmatrix}.$$

The problem in space (4.24), using $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$, can thus be rewritten as:

$$\hat{A}_{\mathcal{S}} \hat{\boldsymbol{\varphi}} = \hat{\boldsymbol{b}}_{\mathcal{S}}, \tag{4.25}$$

with:

$$\hat{A}_{\mathcal{S}} = R^T A_{\mathcal{S}} R = \begin{bmatrix} k_{qq}\hat{\Lambda} + m_{qq} I_r & c_{qp} I_r \\ c_{pq} I_r & m_{pp} I_r \end{bmatrix},$$

$$\hat{\boldsymbol{b}}_{\mathcal{S}} = R^T \boldsymbol{b}_{\mathcal{S}} = \begin{bmatrix} \hat{\boldsymbol{b}}_q \\ \hat{\boldsymbol{b}}_p \end{bmatrix}.$$

The structure of $\hat{A}_{\mathcal{S}}$ exhibits a coupling of the components of $\hat{\boldsymbol{\varphi}}_q$ and $\hat{\boldsymbol{\varphi}}_p$. The solution can be explicitly expressed, component-wise, as:

$$\hat{\boldsymbol{\varphi}}_{q,i} = \frac{1}{\hat{m}_{qq} m_{pp} - c_{qp} c_{pq}} \left(m_{pp} \hat{\boldsymbol{b}}_{q,i} - c_{qp} \hat{\boldsymbol{b}}_{p,i}\right), \tag{4.26}$$

$$\hat{\boldsymbol{\varphi}}_{p,i} = \frac{1}{\hat{m}_{qq} m_{pp} - c_{qp} c_{pq}} \left(\hat{m}_{qq} \hat{\boldsymbol{b}}_{p,i} - c_{pq} \hat{\boldsymbol{b}}_{q,i}\right), \tag{4.27}$$

with:

$$\hat{m}_{qq} = k_{qq} \hat{\lambda}_i + m_{qq}.$$

The complexity of the spatial problem (4.24) is linear in terms of the dimension $r$ of the Ritz subspace, which results in the overall complexity of order $\mathcal{O}\left(mk_{\max}\left(r+n_t+n_\alpha+n_\beta\right)\right)$. It is noteworthy that part of the complexity of the computation is transferred to the solution of the generalized eigenproblem, i.e. $K\boldsymbol{u}=\lambda M\boldsymbol{u}$. However, the solution of the eigenproblem, followed by the PGD offline phase, allows one to obtain the solution to the parametrized problem for all parameter values, in contrast to one FEM solution per pair $(\alpha,\beta)$ for all possible values of $\alpha$ and $\beta$. Strictly speaking, this method consists in searching for the spatial modes as a linear combination of Ritz vectors and can be interpreted as an extra reduction step *per se*. It will be shown to be extremely efficient computation-wise.

### Hybrid space solver

The solution of the projection spatial problem (4.25) involves a trade-off between computational efficiency and solution accuracy. On one hand, the projection of the system onto a subspace spanned by Ritz vectors is extremely efficient but introduces additional numerical errors. Moreover, it was observed that the accuracy of the PGD approximation is bounded by the information contained in the Ritz vectors. On the other hand, solving the full system (4.24) does not introduce any extra approximation and thus enables one to keep finding modes that increase the ROM accuracy. Unfortunately, as it relies on assembling and factorizing the matrix $A_{\mathcal{S}}$ in (4.24) at each fixed-point iteration, this turns out to be prohibitive as soon as the number of spatial degrees of freedom becomes too large. We refer the reader to [100] for further details on this issue.

In this section, we propose a compromise, which consists in adaptively searching for spatial modes either as a linear combination of Ritz vectors, performing a projection (4.25), or as a linear combination of the full-order FE basis functions, solving for the full problem (4.24). In fact, the relevant information contained in the Ritz vectors is usually extracted after around 20 modes (see e.g. Figure 4.13). However, when dealing with a problem involving several parameters, the number of enrichment should be increased to obtain a satisfactory accuracy, in which case one should consider solving the full-order problem (4.24). As in [100], spatial modes $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ are orthonormalized with respect to $K$ and $M^{-1}$, respectively. For the damping parameter modes, they are normalized such that:

$$\begin{aligned}
&\int_{\mathcal{D}_\alpha}\left(\xi_i^q\right)^2 d\alpha=1, \quad \int_{\mathcal{D}_\alpha}\left(\xi_i^p\right)^2 d\alpha=1, \\
&\int_{\mathcal{D}_\beta}\left(\zeta_i^q\right)^2 d\beta=1, \quad \int_{\mathcal{D}_\beta}\left(\zeta_i^p\right)^2 d\beta=1,
\end{aligned} \qquad i=1,\ldots,m.$$

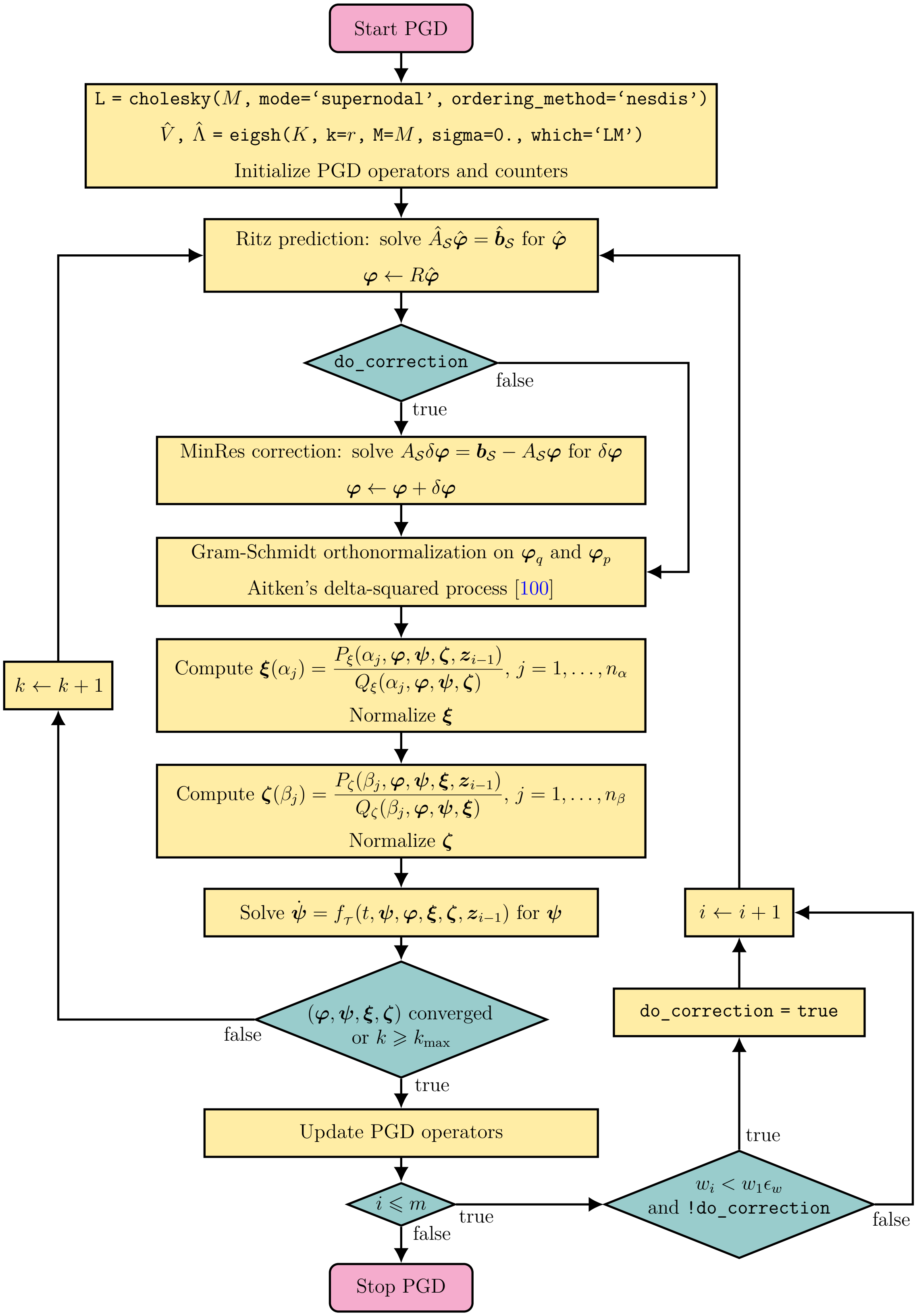

Start PGD
L = cholesky($M$, mode='supernodal', ordering_method='nesdis')
$\hat{V}$, $\hat{\Lambda}$ = eigsh($K$, k=$r$, M=$M$, sigma=0., which='LM')
Initialize PGD operators and counters
Ritz prediction: solve $\hat{A}_{\mathcal{S}}\hat{\boldsymbol{\varphi}} = \hat{\boldsymbol{b}}_{\mathcal{S}}$ for $\hat{\boldsymbol{\varphi}}$
$\boldsymbol{\varphi} \leftarrow R\hat{\boldsymbol{\varphi}}$
do_correction
false
true
MinRes correction: solve $A_{\mathcal{S}}\delta\boldsymbol{\varphi} = \boldsymbol{b}_{\mathcal{S}} - A_{\mathcal{S}}\boldsymbol{\varphi}$ for $\delta\boldsymbol{\varphi}$
$\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + \delta\boldsymbol{\varphi}$
Gram-Schmidt orthonormalization on $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$
Aitken's delta-squared process [100]
$k \leftarrow k+1$
Compute $\boldsymbol{\xi}(\alpha_j) = \dfrac{P_\xi(\alpha_j, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta}, \boldsymbol{z}_{i-1})}{Q_\xi(\alpha_j, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta})}$, $j = 1, \dots, n_\alpha$
Normalize $\boldsymbol{\xi}$
Compute $\boldsymbol{\zeta}(\beta_j) = \dfrac{P_\zeta(\beta_j, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{z}_{i-1})}{Q_\zeta(\beta_j, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi})}$, $j = 1, \dots, n_\beta$
Normalize $\boldsymbol{\zeta}$
Solve $\dot{\boldsymbol{\psi}} = f_\tau(t, \boldsymbol{\psi}, \boldsymbol{\varphi}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{z}_{i-1})$ for $\boldsymbol{\psi}$
$i \leftarrow i+1$
do_correction = true
$(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ converged
or $k \geqslant k_{\max}$
false
true
Update PGD operators
true
$i \leqslant m$
true
false
$w_i < w_1\epsilon_w$
and !do_correction
false
Stop PGD

Figure 4.2 Flowchart of the PGD offline phase with the hybrid space solver.

Thus, for each PGD enrichment, a factor $w_i$ can be defined in the space-discrete formalism as:

$$w_i = \int_{\mathcal{D}} \int_{\mathcal{I}} \left( \frac{1}{2} \left(\psi_i^q \xi_i^q \zeta_i^q\right)^2 \left(\boldsymbol{\varphi}_i^q\right)^T K \boldsymbol{\varphi}_i^q + \frac{1}{2} \left(\psi_i^p \xi_i^p \zeta_i^p\right)^2 \left(\boldsymbol{\varphi}_i^p\right)^T M^{-1} \boldsymbol{\varphi}_i^p \right) dt d\mu.$$

With separability and normalizations, the integral reduces to:

$$w_i = \frac{1}{2} \int_{\mathcal{I}} \left(\psi_i^q\right)^2 + \left(\psi_i^p\right)^2 dt.$$

Alike mode participation factors in modal analysis, $w_i$ measures the contribution of the $i^{\text{th}}$ mode in the description of the dynamic response. We will hereafter refer to these factors as “contribution factors”. Their magnitude is expected to decrease as the rank of the enrichment increases. We thus propose the following adaptive approach (see Figure 4.2):

1. Compute enrichment terms using the Ritz approximation of the problem in space: solve (4.25) for $\hat{\boldsymbol{\varphi}}$ and set $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$. Repeat until $w_i/w_1 < \epsilon_w$, a user-defined tolerance, at which point a MinRes correction will be activated, as described in Step 2;

2. Use the mode $\boldsymbol{\varphi}$ computed in Step 1 as an initial guess $\boldsymbol{\varphi}^0 := \boldsymbol{\varphi}$ and solve (4.24) for $\boldsymbol{\varphi}$ by means of the Minimal Residual (MinRes) iterative solver. This approach is similar to a prediction-correction method.

Such an algorithm circumvents the computational burden incurred by the resolution of (4.24). Furthermore, the prediction of the spatial mode given by $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$ has proved effective when used as an initial guess, since it significantly decreases the number of MinRes iterations (as shown in Figure 4.14).

## 4.3 Parameter identification using Particle Swarm Optimization

The problem we address here deals with the identification of the parameters $\alpha$ and $\beta$ with respect to given snapshots $u_{\mathcal{S}}$. The goal is to find values $(\alpha^*, \beta^*) \in \mathcal{D}$ such that a displacement field, denoted by $u_h = u_h(\alpha, \beta) \in \mathcal{M}$ and evaluated by means of a chosen computational method, minimizes a given cost function $J : \mathcal{D} \to \mathbb{R}$. The problem thus reads:

$$\text{Find } (\alpha^*, \beta^*) \in \mathcal{D} \text{ such that } (\alpha^*, \beta^*) \in \underset{(\alpha,\beta)\in\mathcal{D}}{\operatorname{argmin}} J(\alpha, \beta).$$

The cost function measures the deviation between the snapshot $u_{\mathcal{S}}$ and the computed approximation $u_h$, usually defined with respect to a Quantity of Interest (QoI). In the test case that will be presented thereafter, we focus on the mechanical response over the boundary $\partial\Omega_N$, where Neumann boundary conditions are enforced. The cost function can be formulated as:

$$J(\alpha,\beta) = \frac{\sqrt{\int_{\mathcal{I}}\int_{\partial\Omega_N} \frac{1}{2}\rho\dot{e}(\alpha,\beta)\cdot\dot{e}(\alpha,\beta) + \frac{1}{2}\varepsilon\Big(e(\alpha,\beta)\Big) : \mathbb{E} : \varepsilon\Big(e(\alpha,\beta)\Big)\ dxdt}}{\sqrt{\int_{\mathcal{I}}\int_{\partial\Omega_N} \frac{1}{2}\rho v_s \cdot v_s + \frac{1}{2}\varepsilon\left(u_s\right) : \mathbb{E} : \varepsilon\left(u_s\right)\ dxdt}},$$

with:

$$e(\alpha,\beta) = u_s - u_h(\alpha,\beta),$$
$$\dot{e}(\alpha,\beta) = v_s - \dot{u}_h(\alpha,\beta).$$

The optimization problem is computationally demanding, particularly if $u_h(\alpha,\beta)$ is evaluated by a conventional FEM solver. Thus, the use of the PGD as a surrogate model should improve the efficiency of the process as it provides fast evaluations of $u_h(\alpha,\beta)$ (see Figure 4.3).

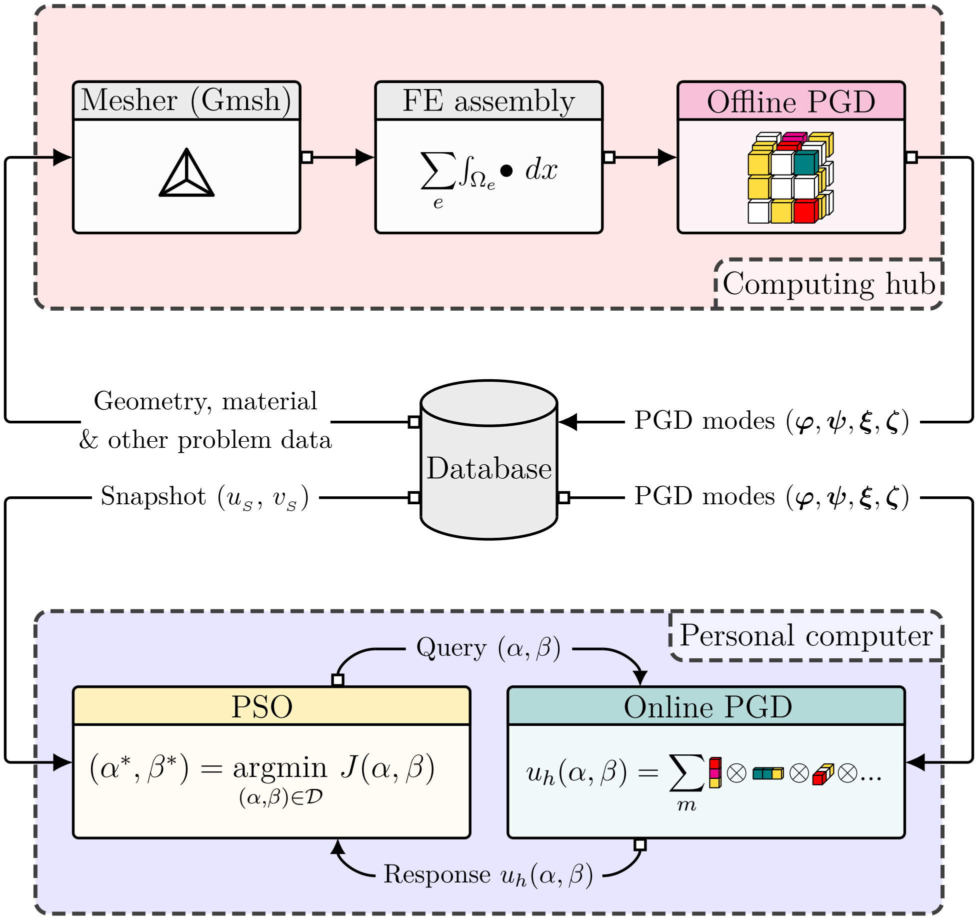

Mesher (Gmsh)
FE assembly
$\sum_e \int_{\Omega_e} \bullet\ dx$
Offline PGD
Computing hub
Geometry, material
& other problem data
Database
PGD modes $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$
Snapshot $(u_s,\ v_s)$
PGD modes $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$
Personal computer
Query $(\alpha,\beta)$
PSO
$(\alpha^*,\beta^*) = \underset{(\alpha,\beta)\in\mathcal{D}}{\text{argmin}}\ J(\alpha,\beta)$
Online PGD
$u_h(\alpha,\beta) = \sum_m \otimes \otimes \otimes ...$
Response $u_h(\alpha,\beta)$

Figure 4.3 Pseudo flowchart of the deployed solution.

The optimization problem is solved using the Particle Swarm Optimization (PSO) technique [101,102], which provides an efficient heuristic that does not require gradient information of the cost function. Therefore, it stands out as a compelling choice when the gradient is not readily available. In this framework, one considers $n_{_{PSO}}$ particles individually associated with a triplet $(\alpha_p, \beta_p, J_p = J(\alpha_p, \beta_p))$, with $p = 1, \ldots, n_{_{PSO}}$, that are used to explore in an iterative manner the search space $\mathcal{D}$. The state of each particle is denoted by $x_p^k = \left(\alpha_p^k, \beta_p^k, J_p^k\right)$, with $k$ being the index of the current PSO iteration and $p$ the particle label. The PSO technique proceeds as follows:

1. Initialization: a population of $n_{_{PSO}}$ particles is randomly initialized within $\mathcal{D}$. Each particle is assigned a triplet $x_p^0 = \left(\alpha_p^0, \beta_p^0, J_p^0\right)$;

2. Individual update: each particle updates its own best state $x_p^{k_p}$, where $k_p = \underset{1 \leqslant i \leqslant k}{\operatorname{argmin}} \, J_p^i$, for $p = 1, \ldots, n_{_{PSO}}$;

3. Global update: update the global best state $x_{p_b}^k$, which represents the best solution found by any particle in the entire swarm at the current iteration, where $p_b = \underset{1 \leqslant p \leqslant n_{_{PSO}}}{\operatorname{argmin}} \, J_p^k$;

4. State update: the state of each particle is updated based on its current state, own best state, and global best state, according to the following equations:

$$\text{Velocity update} \quad \begin{cases} \dot{\alpha}_p^{k+1} = w\dot{\alpha}_p^{k+1} + c_1\epsilon_p^k \left(\alpha_p^{k_p} - \alpha_p^k\right) + c_2\eta_p^k \left(\alpha_{p_b}^k - \alpha_p^k\right), \\ \dot{\beta}_p^{k+1} = w\dot{\beta}_p^{k+1} + c_1\epsilon_p^k \left(\beta_p^{k_p} - \beta_p^k\right) + c_2\eta_p^k \left(\beta_{p_b}^k - \beta_p^k\right), \end{cases}$$

$$\text{Position update} \quad \begin{cases} \alpha_p^{k+1} = \alpha_p^k + \dot{\alpha}_p^{k+1}, \\ \beta_p^{k+1} = \beta_p^k + \dot{\beta}_p^{k+1}, \end{cases}$$

where $w$, $c_1$, and $c_2$ are three control parameters, namely the inertia weight, the cognitive, and the social acceleration coefficients, respectively, and $\epsilon_p^k$ and $\eta_p^k$ are random perturbations. These parameters influence the search strategy between exploitation and exploration. The term exploitation means that one carries out the search around known promising states (own and global best states, promoted by $c_1$ and $c_2$). A drawback of assigning too much weight to exploitation is the premature attraction to local optima. Conversely, exploration promotes the evaluation of states as widely as possible within the search space (promoted by $w$, $\epsilon_p^k$, and $\eta_p^k$). It may in return lead to a higher number of iterations for the algorithm to converge towards an optimal solution. Consequently, the parameters are to be tuned in order to find a good trade-off between exploitation and exploration. Steps 2 to 4 are repeated until a convergence criterion is fulfilled. It is worth noting that evaluating the state of each particle can be seamlessly performed in parallel.

## 4.4 Numerical examples

### 4.4.1 Test case: parameter optimization with respect to a snapshot

The test case consists of a 3D beam clamped on one end $\partial\Omega_D$ and subjected to a multiaxial load on the other end $\partial\Omega_N$, see Figure 4.4. The domain $\Omega = (0,6) \times (0,1) \times (0,1)$ (in meters) is a rectangular parallelepiped with a squared cross-section. Its response to an external load on its right end is computed over the time interval $\mathcal{I} = (0,1)$ (in seconds). The governing equations are those introduced in Section 4.1.1 with $f = 0$. Moreover, the beam is subjected to homogeneous initial conditions:

$$\begin{aligned} u(x,0) &= 0, && \forall x \in \Omega, \\ \frac{\partial u}{\partial t}(x,0) &= 0, && \forall x \in \Omega, \end{aligned}$$

and to the boundary conditions:

$$\begin{aligned} u(x,t) &= 0, && \forall (x,t) \in \partial\Omega_D \times \mathcal{I}, \\ \sigma(u)\cdot n &= g_N(x,t), && \forall (x,t) \in \partial\Omega_N \times \mathcal{I}, \\ \sigma(u)\cdot n &= 0, && \forall (x,t) \in \partial\Omega_0 \times \mathcal{I}. \end{aligned}$$

The external load $g_N$, applied to the boundary $\partial\Omega_N = \{6\} \times (0,1) \times (0,1)$ of the structure, is described in Figure 4.5 and results in both vertical bending and torsion. The beam is free on the remainder of the boundary $\partial\Omega_0 = \partial\Omega\backslash(\partial\Omega_D \cup \partial\Omega_N)$. The values of the parameters are chosen as follows:

$$\begin{aligned} &E = 220 \text{ GPa}, \\ &\nu = 0.3, \\ &\rho = 7{,}000 \text{ kg/m}^3, \\ &\omega_B = 160 \text{ rad/s}, \end{aligned}$$

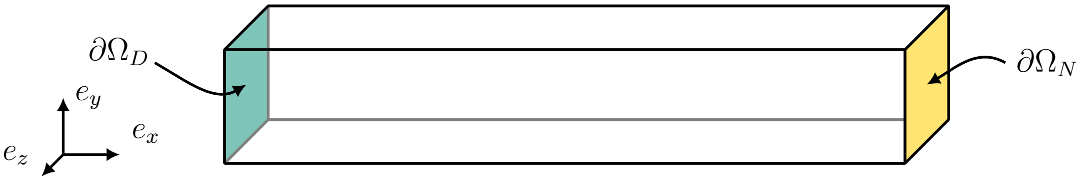


Figure 4.4 Schematic of the 3D beam $\bar{\Omega} = [0,6] \times [0,1] \times [0,1]$.

$$\begin{aligned}
&\omega_T = 875 \text{ rad/s},\\
&\bar{\alpha} = 10^{-5} \text{ s},\\
&\bar{\beta} = 1 \text{ s}^{-1},\\
&\alpha_{\min} = \beta_{\min} = 0.2,\\
&\alpha_{\max} = \beta_{\max} = 2,
\end{aligned}$$

and the Lamé coefficients are evaluated as:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

The time domain $\mathcal{I}$ is divided into $n_t = 4{,}800$ sub-intervals of equal size. The space domain $\Omega$ is partitioned into linear tetrahedral elements and five meshes will be considered such that the number $2n$ of spatial DOFs takes values in $\{33{,}990, 147{,}174, 554{,}412, 1{,}408{,}002, 3{,}002{,}406\}$. Regardless of the spatial discretization, the number of Ritz vectors is set to $r = 300$, corresponding to the smallest Ritz values [15]. The *offline phase* of the PGD is performed for $m = 100$ modes. In the case of the hybrid space solver, the tolerance $\epsilon_w$ is set to $10^{-6}$. The evaluation of the cost function $J$ is carried out using the PGD reduced-order model, referred to as the *online phase*. Unless stated otherwise, $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ are both discretized using $n_\alpha = n_\beta = 190$ values for $\alpha$ and $\beta$. Thus, the dimension of the discrete global parameter space $\mathcal{D}$ is 36,100. Note that the evaluation of $J$ is not limited to these discrete parameter values as it can be evaluated for any pair $(\alpha, \beta) \in \mathcal{D}$ by interpolation of the PGD damping modes $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$. A linear interpolation will be considered for this purpose. The global best PSO variant is used with the inertia weight, the cognitive and the social acceleration coefficients set to $w = 0.4$, $c_1 = 0.5$, and $c_2 = 0.3$, respectively. The PSO algorithm will run in parallel with $n_{PSO} = 32$ particles. Recall that the overall solution process is illustrated in Figure 4.3.

### 4.4.2 Comparison method and performance criteria

We shall report and compare the results based on the following criteria:

1. The relative error $\epsilon_{ROM}$ of the PGD approximations with respect to the full-order solutions, namely the FEM solutions described in Section 4.1.3, defined as:

$$\epsilon_{ROM} = \sqrt{\frac{\|u_{FEM} - u_{ROM}\|_{\mathcal{M}}}{\|u_{FEM}\|_{\mathcal{M}}}}.$$

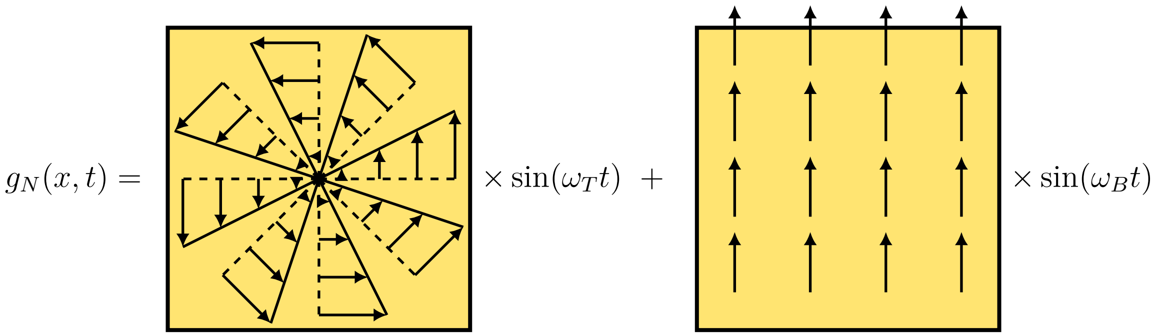


Figure 4.5 Schematic of the boundary multiaxial load $\sigma \cdot n$ prescribed on $\partial\Omega_N$.

For feasibility reasons, $\epsilon_{ROM}$ will be assessed only for a small number of spatial DOF, i.e. $2n = 36{,}774$, and coarse parameter discretizations $n_\alpha = n_\beta = 10$ for $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$. Three approaches will be compared for the problem in space: 1) only the Ritz prediction is performed; 2) an LU factorization is performed at each fixed-point iteration; 3) a prediction-correction with a MinRes solver is performed, as described in Section 4.2.2. The three approaches are respectively labeled "Ritz", "Full", and "Hybrid" on Figure 4.13. The evolution of the contribution factors $w_i$ will be measured to verify the efficiency of the hybrid solver. Past these results, only the hybrid solver will be considered;

2. The time speedup factor achieved by the reduced-order model when compared to the conventional FEM solver, see Algorithm 4. It will include scaling performance with respect to the size of the spatial discretization. The scaling performance refers to the offline phase that was performed on a computing hub offering more resources than most personal computers;

3. The reduction in the cost function evaluated by the swarm particles. The relative errors in the optimized parameters $(\alpha^k_{p_b}, \beta^k_{p_b})$, with respect to the pair $(\alpha_{\scriptscriptstyle S}, \beta_{\scriptscriptstyle S})$ employed to generate the snapshots, are defined as:

$$\epsilon^k_\alpha = \frac{\alpha^k_{p_b} - \alpha_{\scriptscriptstyle S}}{\alpha_{\scriptscriptstyle S}}, \quad \epsilon^k_\beta = \frac{\beta^k_{p_b} - \beta_{\scriptscriptstyle S}}{\beta_{\scriptscriptstyle S}}.$$

   The errors are assessed at every PSO iteration, denoted here by $k$, for the global best particle. The snapshots and the offline PGD are computed on the same mesh featuring $2n = 3{,}002{,}406$ spatial DOFs. The optimization was performed on a personal computer, using the PGD as a surrogate model (online phase) and the PSO to drive the parameter search.

As far as computer times are concerned, the configurations of both the computing hub and the personal computer are detailed below:

- Computing hub (AWS' `c5.12xlarge` instance):
  - Intel(R) Xeon(R) Platinum 8275CL CPU @ 3.00GHz (24 cores, 48 threads);
  - RAM: 96 GB;
  - OS: Ubuntu 22.04.4 LTS (Jammy Jellyfish).
- Personal computer:
  - CPU: AMD Ryzen 7 PRO 4750U @ 1.7 GHz per core (8 cores, 16 threads);
  - RAM: 38 GB;
  - OS: Arch Linux kernel version 6.10.6.

The code, written using Python 3.9.19, leverages Intel's MKL with the Intel Distribution for Python and SuiteSparse's CHOLMOD Supernodal Sparse Cholesky Factorization [98]. The MinRes algorithm is SciPy `scipy.sparse.linalg.minres` [97] with a tolerance for the relative residual set to $\texttt{tol} = 5 \times 10^{-8}$. The offline PGD computation was performed using the computing hub on 16 cores. The PSO, fed with the PGD surrogate model, was carried out by the research toolkit PySwarms [103] using the personal computer on 8 cores.

### 4.4.3 Numerical results

We show in Figure 4.13 the evolution of the relative error $\epsilon_{ROM}$ with respect to the number of modes, when using the Ritz projection, the full, and the hybrid approaches. We observe that $\epsilon_{ROM}$ decreases rather fast for about the first 30 enrichments. However, the limitation of the Ritz projection is clearly highlighted on this test case as we can see that the useful information contained in the $r = 300$ Ritz vectors has been thoroughly exploited after 30 enrichments. The adaptive hybrid solver activates the MinRes correction around the $30^{\text{th}}$ enrichment, after which an abrupt decrease in $\epsilon_{ROM}$ along with an increase of the contribution factors $w_i$ are observed. One aspect that has not been discussed so far is the loss of information due to the projection in Eq. (4.25) on the representation of the original right-hand side of Eq. (4.24). In fact, it is rather difficult to predict in advance up to which rank the representation of the right-hand side residual in the base formed by the Ritz vectors will be accurate. In particular, it is likely that high-frequency variations in the residual will not be retained after projection. This information is however recovered as soon as the hybrid solver switches to the solution of the full-order spatial problem (4.24). Figure 4.13 also shows the error reduction of the PGD

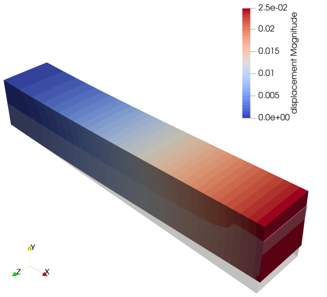


Figure 4.6 1st spatial mode.

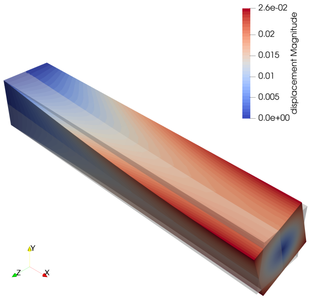


Figure 4.7 2nd spatial mode.

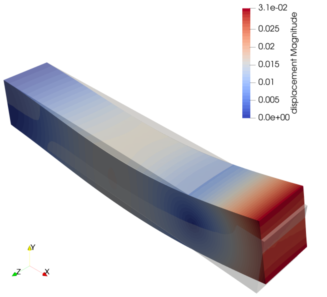


Figure 4.8 3rd spatial mode.

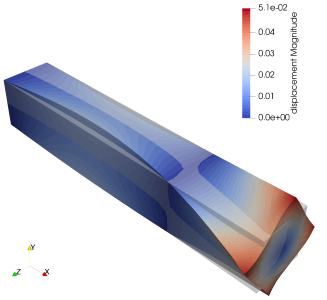


Figure 4.9 4th spatial mode.

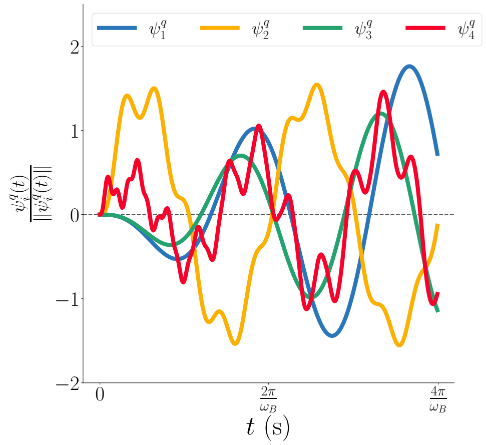


Figure 4.10 First four temporal modes.

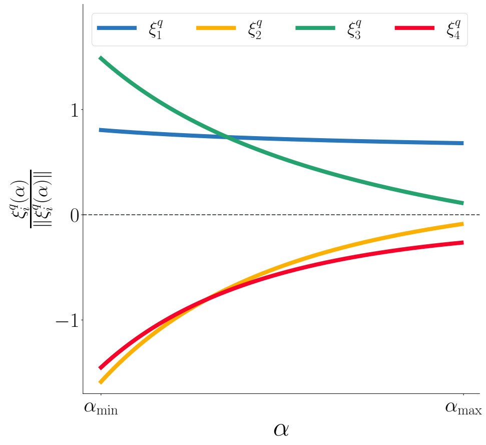


Figure 4.11 First four $\alpha$-modes.

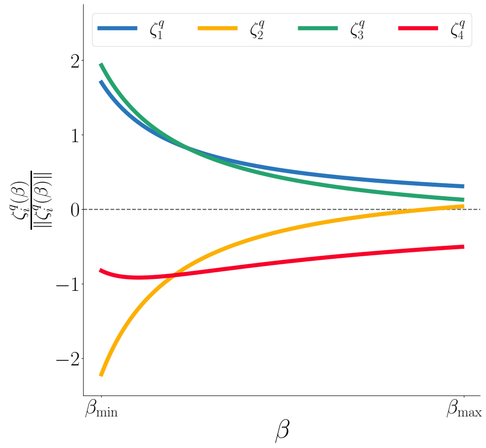


Figure 4.12 First four $\beta$-modes.

solver with LU factorization, labeled “Full”, and it confirms the relevance of the hybrid solver. Indeed, the hybrid solver mitigates computational costs with an accuracy similar to that of the “Full” solver. As a point of comparison, for 33,990 spatial DOFs, the offline phase takes 970 seconds when an LU factorization is used to solve the problem in space, compared to 24 seconds with the hybrid strategy. Regarding the latter, Figure 4.14 illustrates the impact of the prediction-correction scheme, reducing the average number of MinRes iterations per linear solve (4.24) from 91 to 20. It is also worth mentioning that the Cholesky factorization of $K$ was tested as a preconditioner for the MinRes solver, but this choice was not retained. While it reduces the number of iterations to an average of 10, the additional computations required for the Cholesky forward and backward solves ultimately make this approach slower.

The results about execution time are reported in Tables 4.1 and 4.2. Note that the execution time for the FEM solver is that of a computation for a single pair $(\alpha, \beta)$. As multiple evaluations of the response are needed to proceed with parameter optimization, one may conclude that the use of a classic FEM solver is prohibitive. Conversely, once the PGD offline phase is completed, the modes can be used as a surrogate model to quickly assess the mechanical behavior for any pair $(\alpha, \beta) \in \mathcal{D}$. Figure 4.15 shows the time distribution across the different phases of the computations. It clearly illustrates that the time marching scheme is the computationally most expensive phase of the FEM solver. In the case of the PGD solver, the space problem accounts for the majority of the computational time. The diagram clearly demonstrates the excellent efficiency of the Ritz estimates, even when including the time spent to compute the Ritz vectors during the preprocessing stage. Notably, the damping modes contribute minimally to the overall computational time. Note that the time required to assemble the operators is included in the diagram slices for each problem, respectively. Therefore, a possible improvement would be to work on a more efficient method for assembling the operators for the right-hand side residuals and the left-hand side (when applicable). This remark especially applies to the problem in space. Indeed, the problems related to time and damping parameters are purely one-dimensional with respect to the time variable $t$ and damping parameters $\alpha$ and $\beta$. In contrast, the spatial problem does not assume separability with respect to the three spatial directions, which results in an increased workload.

Figure 4.16 shows the evolution of the cost function evaluated for both the global best and global worst particles of the swarm as well as the relative errors $\epsilon_\alpha^k$ and $\epsilon_\beta^k$ for the global best particle. We illustrate in Figure 4.17 the PSO iterative search and in Figure 4.18 the surface of the cost function near the global minimum. Note that the surface was post-processed for visualization purposes only as PSO does not require any knowledge other than the one collected by the swarm particles through their search. One observes in Figure 4.16 that the best particle reaches an optimal state around the 40th iteration. The convergence of

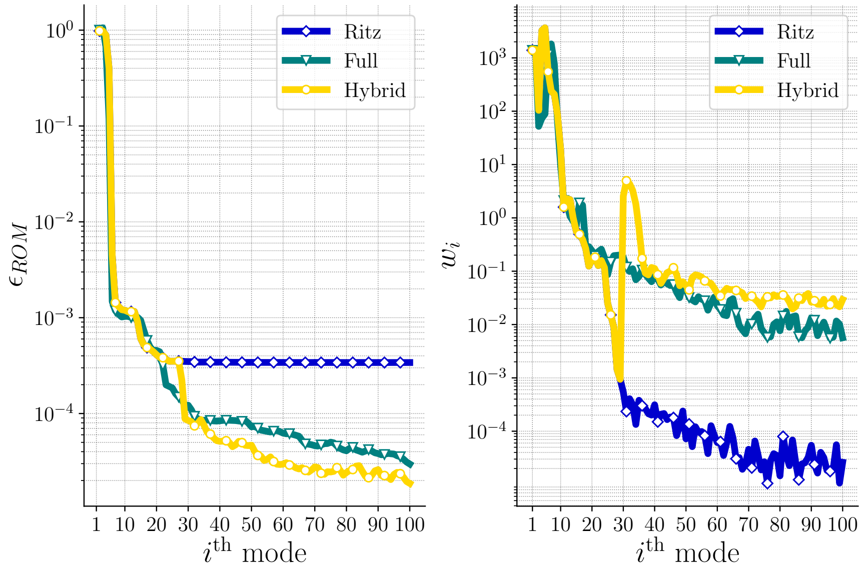


Figure 4.13 (Left) Relative error $\epsilon_{ROM}$ between the reference fields and the PGD fields for the three strategies for the problem in space ($y$-axis in log scale). (Right) Factor $w_i$ of the $i^{\text{th}}$ enrichment for the three strategies for the problem in space ($y$-axis in log scale).

the swarm towards this optimum is reached around the 55th iteration. The final optimized damping coefficients achieve errors that are within acceptable bounds, and the value of the optimized cost function also indicates the effectiveness of the approach. Thus, we consider that $55 \times n_{PSO} = 1{,}760$ evaluations of the cost function, and a fortiori of $u_h(\alpha, \beta)$, were required to solve the optimization problem. Using the PGD surrogate model (online phase), the optimization was performed in 322 seconds. If a conventional FEM solver were to be used instead, this time would have been several orders of magnitude longer. Accounting for the offline phase, the overall speedup factor is estimated to be 3,800. Beyond time efficiency,

Table 4.1 Time efficiency of the PGD offline phase with respect to different spatial discretizations with $n_\alpha = n_\beta = 190$.

| **# DOF in space** | 33,990 | 147,174 | 554,412 | 1,408,002 | 3,002,406 |
|---|---|---|---|---|---|
| **Offline PGD (s)** | 24 | 90 | 360 | 999 | 2,317 |

Table 4.2 Time efficiency of the FEM solver described in Algorithm 4 for one computation.

| **# DOF in space** | 33,990 | 147,174 | 554,412 | 1,408,002 | 3,002,406 |
|---|---|---|---|---|---|
| **FEM (s)** | 29 | 188 | 847 | 2,431 | 5,757 |

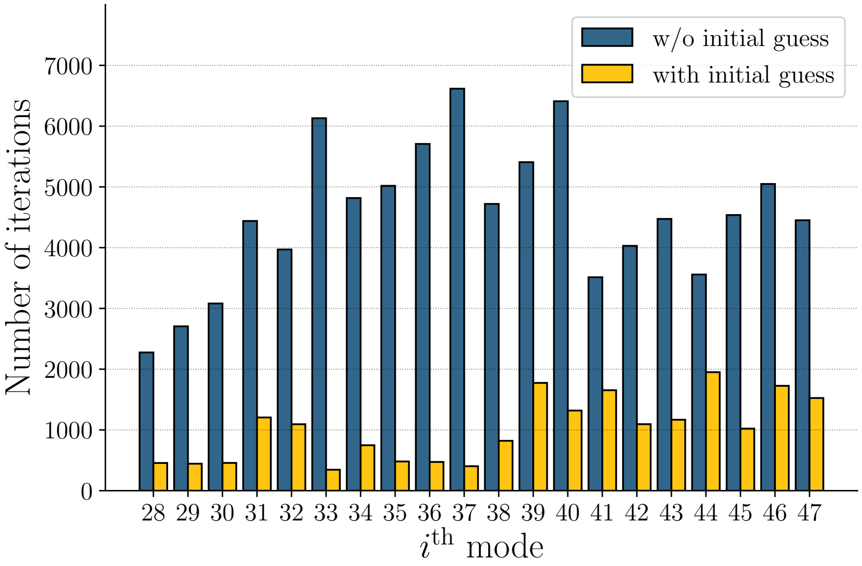


Figure 4.14 Impact of an initial guess on the number of MinRes iterations (cumulative sum of MinRes iterations per fixed-point iteration).

the memory usage needed to store the results is also an aspect to be accounted for. After the offline phase, we are provided with separated representations for both the generalized coordinates and momenta, i.e. $2m$ modes for the space, time, and damping parameters, respectively, which amount to a total of $2m(n + n_t + n_\alpha + n_\beta)$ floating numbers to store. By contrast, the storage of all FEM solutions amounts to $2n \times n_t \times n_\alpha \times n_\beta$ floating numbers. Therefore, the PGD approximation also offers a gain in memory by several orders of magnitude. Table 4.3 summarizes these results, considering that numbers are stored in double precision. From a computational point of view, the PGD offline phase may be seen as a substantial overhead with respect to the optimization problem. However, the computational cost of such an approach is subsequently justified by the gains it enables.

A last noteworthy comment is about the physical interpretation of the results. The graphical representation of the parameter-separated modes provides a picture of the influence of each parameter used to simulate the physical phenomena. As a reminder, our simulation consists of a viscoelastic 3D beam that is clamped on one end and subjected to bending and torsional, periodic, mechanical efforts on the other end. The bending and torsional loading functions have different pulsations, denoted by $\omega_B$ and $\omega_T$, respectively. We observe in Figures 4.6–4.12 that the computed PGD modes are rather consistent with the physics being modeled. Indeed,

1. The spatial modes $\{\boldsymbol{\varphi}_1^q, \boldsymbol{\varphi}_3^q\}$, shown in Figures 4.6 and 4.8, and $\{\boldsymbol{\varphi}_2^q, \boldsymbol{\varphi}_4^q\}$, shown in

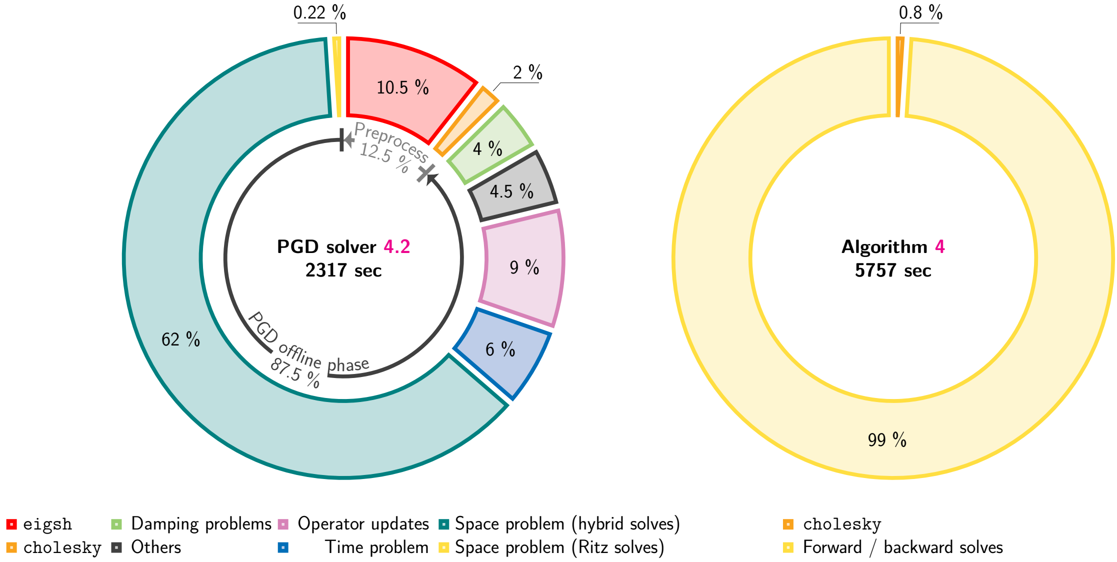


Figure 4.15 (Left) Time distribution of the developed PGD solver described on Figure 4.2 ($2n$ = 3,002,406 spatial DOFs, $n_t$ = 4,800 time-steps and $n_\alpha = n_\beta$ = 190 values for the damping parameters). (Right) Time distribution of the FEM solver described in Algorithm 4 ($2n$ = 3,002,406 spatial DOFs, $n_t$ = 4,800 time-steps for one computation).

Figures 4.7 and 4.9, contribute to the bending and torsion motions, respectively;

2. The temporal modes $\{\psi_1^q, \psi_3^q\}$, see Figure 4.10, are clearly related to the bending frequency $\omega_B$, while the modes $\{\psi_2^q, \psi_4^q\}$ are similar to signals featuring two frequencies. The slow and fast variations seem to be related to $\omega_B$ and $\omega_T$, respectively. For mode $\psi_4^q$, some higher frequencies are visible in the vicinity of $t = 0$. We interpret them as small corrections as they do not appear to have a particular physical meaning regarding the model;

3. Damping modes: their magnitudes tend to decrease as the values of the damping parameters increase, which is physically sensible.

### 4.4.4 Further discussion

In this section, we make several comments in relation to the proposed PGD strategy.

We first recall that the estimation of time and memory for the optimization phase with a conventional FEM solver, as reported in Table 4.3, was based on the fact that one solution was computed at a time, the solution fields were only stored over $\partial\Omega_N \times \bar{\mathcal{I}}$, and the data was erased after the cost function was evaluated. Tackling an optimization problem this way is an energy drain. Conversely, the PGD allows to store once and for all the solution fields

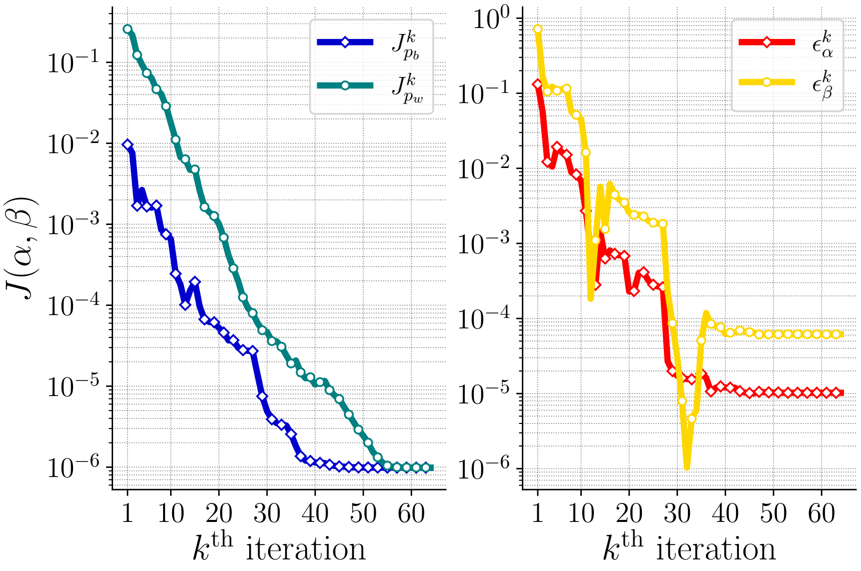


Figure 4.16 (Left) Values of the global best cost $J^k_{p_b}$ and the global worst cost $J^k_{p_w}$ during the PSO iterations. (Right) Values of the errors $\epsilon_\alpha$ and $\epsilon_\beta$ for the global best particle $p_b$ during the PSO iterations.

over $\bar{\Omega} \times \bar{\mathcal{I}} \times \mathcal{D}_\alpha \times \mathcal{D}_\beta$. The data compression enabled by the PGD allows one to reuse past computations to carry out new analyses. For instance, if one wanted to study a specific QoI, e.g. the dissipated energy, the solution is readily available and one could simply skip the offline phase.

The proposed PGD solver begins with the computation of eigen-pairs approximations, as in Modal Decomposition. The mechanical behavior of the structure could be efficiently assessed for several damping parameters using modal superposition instead of PGD. However, as previously discussed in [100], the drawback of the method is that many of the Ritz vectors are irrelevant to describe the behavior of the structure, since they do not account for external loads. We actually computed that, for $2n = 33{,}990$ spatial DOFs, the computation of around 16,000 Ritz vectors (approximately 4 hours on the computing hub) was needed to recover the same accuracy as that provided by the PGD approach using 100 modes. This shortcoming in the modal superposition method can be mitigated by considering additional modes, referred to in the literature as static corrections [15]. In that regard, it also seems interesting to enrich the subspace spanned by the Ritz vectors with such static corrections in the context of the PGD approach presented herein.

Finally, it is important to highlight that the success of PGD depends on the separability of the input data. This is indeed the case here: the external load is explicitly expressed as a sum of products of space and time functions (recall Figure 4.5), and the damping operator

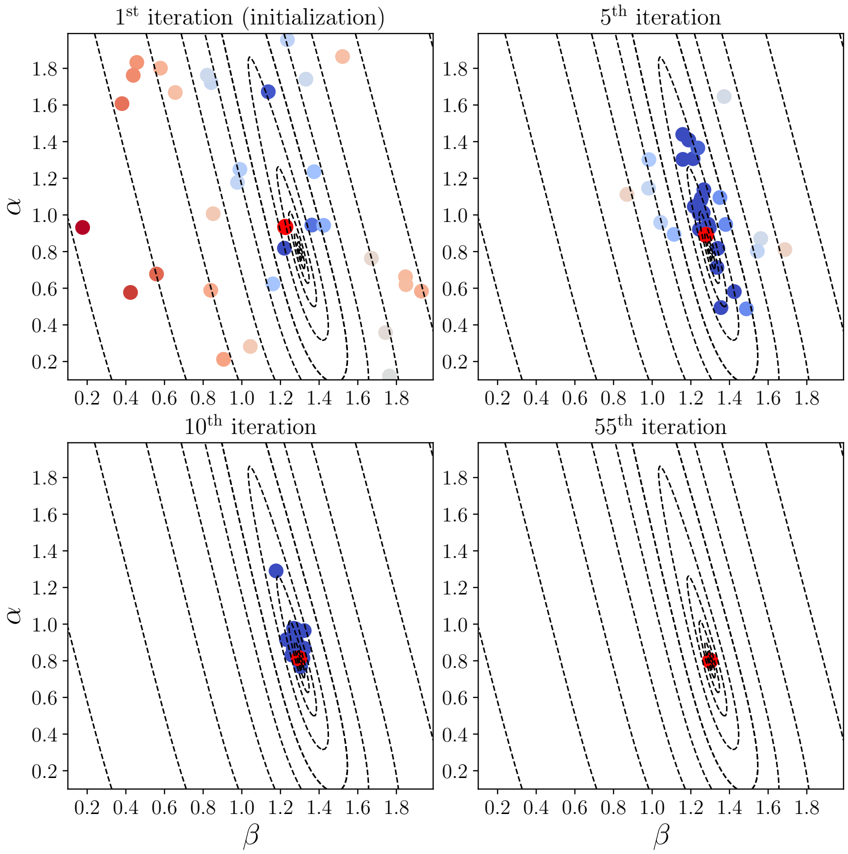


Figure 4.17 The swarm particles throughout PSO iterations.

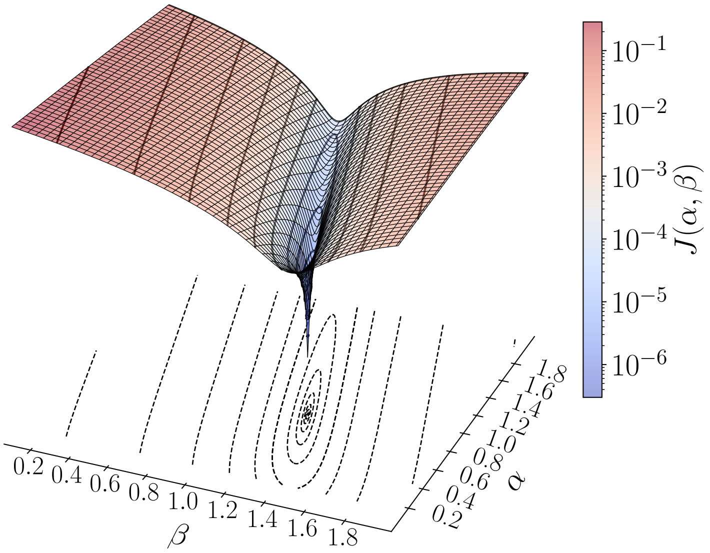


Figure 4.18 Cost function $J$ over $\mathcal{D}$.

Table 4.3 Summary of the time and memory efficiency of the developed PGD solver against those of a conventional FEM for a computation with $2n$ = 3,002,406 spatial DOFs, $n_t$ = 4,800 time-steps and $n_\alpha = n_\beta = 190$ values for the damping parameters (the PGD results are provided by the code while the FEM results are estimated based on the result for a single computation).

| | | **PGD** | **FEM** |
|---|---|---|---|
| **Offline phase** | time (s) | 2,317 | n.a. |
| | memory (GB) | 59 | n.a. |
| **Optimization** | time (s) | 322 | $10^7$ |
| | memory (GB) | 2.7 | 52 |
| **Storage** | | 4.8 GB | 4,200 TB |

is written as a sum of products of the damping parameters with the stiffness and mass matrices. However, if the input data is not explicitly separable, one can approximate a separated representation using the SVD or, in the case of geometric parameters for example, employ morphing techniques with respect to a reference mesh [66]. In general, while these tasks are essential to the performance of PGD-based solvers, they are not trivial and can be computationally intensive.

## 4.5 Conclusion

We have presented in this chapter an extended implementation of the PGD approach to efficiently parametrize reduced-order models with respect to Rayleigh damping coefficients. The results demonstrated that the hybrid solver, based on a two-step prediction-correction process, successfully mitigated computational costs while achieving accuracy comparable to that of full-order solutions. The spatial modes were initially estimated in the subspace formed by the Ritz vectors, as long as the contribution of these modes is significant. When this was no longer the case, the estimates were further refined using a Minimal Residual iterative solver, thereby capturing additional significant information. This hybrid strategy effectively balanced the trade-off between computational expense and solution accuracy. Ultimately, the PGD reduced-basis was integrated as a surrogate in the Particle Swarm Optimization algorithm to optimize the damping parameters for a given snapshot, demonstrating the effectiveness of the proposed PGD approach for optimization. The numerical experiments confirmed that the parametric PGD can accurately build a ROM that includes damping modes and efficiently identifies the damping coefficients used in the snapshot.

# CHAPTER 5 CONCLUSION

## 5.1 Summary of Works

We have presented in this thesis some novel methodologies and algorithms to enhance the performance of the PGD framework for structural dynamics, with a particular emphasis on applications dealing with the simulation of viscoelastic material subjected to transient external loads. Focusing on computational efficiency and structure preservation, the PGD framework has been revisited using the Hamiltonian formalism and was further extended to incorporate damping parameters as variables of the reduced model.

Initially, the robustness issue of such methods was addressed by developing a PGD solver based on a displacement-momentum formulation. This approach simultaneously builds two bases for the displacement and conjugate momentum, allowing for the orthogonalization of modes using well-chosen metrics to prevent ill-conditioning and divergence. The PGD reduced model was further extended to a space-discrete, time-continuous Hamiltonian formalism, ensuring the preservation of the symplectic structure in time. Computational costs are mitigated by introducing an approximation inspired by Modal Decomposition for solving spatial problems. Specifically, the spatial modes are computed in the subspace formed by the Ritz vectors, which approximate the eigenvectors of the structure, rather than in the basis of the finite element basis functions. Additionally, Aitken's delta-squared process and mode-orthogonalization are incorporated into the fixed-point iteration procedure to enhance convergence and stability.

Building on these developments, the PGD framework was finally extended to efficiently parametrize the reduced-order model with respect to Rayleigh damping coefficients. The approach combines a two-step prediction-correction process, using first the Ritz vectors for initial estimates, and then employing a Minimal Residual iterative solver to improve accuracy in the spatial solutions. This hybrid strategy balances computational efficiency and solution precision. The resulting PGD reduced basis was subsequently employed within a Particle Swarm Optimization algorithm as a surrogate to determine optimal damping coefficients based on a given snapshot.

Numerous numerical experiments have been presented to illustrate the performance of the proposed methodologies. They have demonstrated that the PGD framework could tackle large-scale problems while preserving the original symplectic structure or providing a damped surrogate model.

### 5.2 Limitations and Future Research

The limitations of PGD mainly stem from the validity of the separability assumption. It is particularly challenging to employ this method in cases where the input data cannot be easily separated. In the present study, the input data that was prescribed was readily expressed in a separated format. However, one practical example where such an issue could arise is the parametrization of periodic external loads with respect to their component frequencies, i.e. when $f$ is assumed in the form:

$$f(x,t,\boldsymbol{\omega}) = \sum_{i=1}^{p} f_i(x)\sin(\omega_i t),$$

with $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_p]$ being a vector of $p$ pulsations that are considered as parameters of the ROM. While the separation with respect to spatial variables is straightforward, it becomes less trivial when dealing with the time $t$ and pulsations $\omega_i$. In such cases, some strategies where partial separation is applied may come in handy. It would consist in assuming the solution fields can be written as:

$$u(x,t,\boldsymbol{\omega}) = \sum_{i=1}^{r} \varphi_i(x)\psi_i(t,\boldsymbol{\omega}),$$

where the unknown modes $\psi_i$ would be computed by means of a dedicated solver. The design of such PGD solvers with partial separation on extra parameters and their efficiency would have to be studied on a case-by-case basis. Another shortcoming comes directly from the model itself, with the assumptions of both viscoelastic materials and infinitesimal strains. It would be interesting to explore the potential of the presented methodologies in the case of nonlinear problems.

Although the presented work has led to a substantial improvement in efficiency, it was observed that the problem in space remains a major bottleneck. On one hand, the spatial problem does not assume separability with respect to the three spatial directions, which is a strong assumption, that is, solution fields are sought in the form:

$$u(x,y,z,t,\alpha,\beta) = \sum_{i=1}^{r} \varphi_i(x,y,z)\psi_i(t)\xi_i(\alpha)\zeta_i(\beta),$$

where the dependence on the three spatial coordinates $(x,y,z)$ is explicitly written. On the other hand, subproblems related to time and damping parameters are purely one-dimensional with respect to the time variable $t$ and damping parameters $\alpha$ and $\beta$. As a result, a tangible workload difference was observed between the PGD subproblems. The Ritz projection and hybrid approaches drastically improved the efficiency of the PGD solver. However, this improvement came at the cost of the computation of Ritz pairs, i.e. approximated solutions of the generalized eigenproblem $K\boldsymbol{u} = \lambda M\boldsymbol{u}$. A refined approach could involve eliminating the need to solve the eigenproblem by employing augmented Krylov-based solvers.

An augmented Krylov-based solver becomes highly effective in addressing the observed limitations, particularly by integrating techniques such as deflation and recycling with preconditioning [93]. Deflation reduces the dimensionality of the spatial problem by subtracting the contribution of previously computed PGD spatial modes. This is especially beneficial when earlier modes capture dominant behaviors in the spatial domain. In practice, deflation ensures that the known modes do not contribute to the residual in future iterations. This is done by projecting the residual onto the orthogonal complement of the subspace spanned by the known modes. As a result, the solver's search directions are orthogonal to the previously computed modes, effectively speeding up convergence. In the proposed work, this deflation was applied in the fixed-point iteration at line 7 in Algorithm 3, rather than being integrated directly into the linear solver. The recycling process further enhances efficiency by reusing Krylov subspaces generated from previous PGD iterations, avoiding the need to reconstruct the subspace from scratch. Additionally, preconditioning can improve convergence rates, although the choice of the preconditioner would require an in-depth study.

Furthermore, while `Python` provides a modular and user-friendly development environment, it would be interesting to consider the implementation using a lower-level language such as `C++`. It would provide greater control over system resources and would significantly enhance computational efficiency. Offloading the treatment of the spatial problem to the GPU may also be considered. GPU offloading would be particularly beneficial for tasks related to the problem in space (spatial residual updates and spatial solves). However, a successful implementation requires careful software design, including strategies to keep spatial operators GPU-resident while other operators remain CPU-resident. Effective management of these aspects is essential to maximize the benefits of GPU offloading.

The use of a symplectic time integrator along with the symplecticity preservation in the spatial modes could also provide a strong foundation for extending this work. This may enable the development of a reduction technique suited to elastodynamics problems involving large rotations and small strains, as discussed in [39]. Another avenue to improve efficiency and accuracy could be to include error estimation and adaptivity to construct PGD approximations tailored to the prediction of quantities of interest, as in [99].

Finally, with the dramatic increase in interest regarding Machine Learning, it is worth noting that a Separable DeepONet (SepDeepONet) framework has been recently proposed [104].

Like the PGD, SepDeepONet breaks the problem down by handling one-dimensional coordinates individually, with each sub-network dedicated to computing the separated modes for its respective coordinate. As a result, it reduces both the number of forward passes and the computational burden of Jacobian matrix calculations. This study has demonstrated that the SepDeepONet framework also achieves linear scaling of computational cost with increasing discretization density. This new framework, and Machine Learning methods more generally, could be a promising area for future research.

# REFERENCES

[1] S. Edwards and D. Harris, *Hidden human computers: The black women of NASA*, ser. Hidden heroes. Minneapolis, MN: Essential Library, an imprint of Abdo Publishing, 2017. Cited on Page 1.

[2] J. T. Oden, T. Belytschko, I. Babuska, and T. J. R. Hughes, "Research directions in computational mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 913–922, 2003. Cited on Pages 1 and 2.

[3] É. de France, "Finite element ***code_aster***, Analysis of Structures and Thermomechanics for Studies and Research," Open source at https://www.code-aster.org, 1989–2024. Cited on Page 2.

[4] D. Arndt, W. Bangerth, M. Bergbauer, M. Feder, M. Fehling, J. Heinz, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, B. Turcksin, D. Wells, and S. Zampini, "The `deal.II` Library, Version 9.5," *Journal of Numerical Mathematics*, vol. 31, no. 3, pp. 231–246, 2023. Cited on Page 2.

[5] M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. N. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, "The FEniCS Project Version 1.5," *Archive of Numerical Software*, vol. 3, 2015. Cited on Page 2.

[6] F. Hecht, "New development in FreeFem++," *J. Numer. Math.*, vol. 20, no. 3-4, pp. 251–265, 2012. Cited on Page 2.

[7] L. Chamoin and P. M. B. Díez, "Verifying calculations - forty years on: An overview of classical verification techniques for fem simulations," 2015. Cited on Page 2.

[8] P. Benner, S. Gugercin, and K. E. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM Rev.*, vol. 57, pp. 483–531, 2015. Cited on Pages 2, 11, 15, and 105.

[9] R. Bellman, "Dynamic programming," *Science*, vol. 153, pp. 34 – 37, 1957. Cited on Page 3.

[10] B. Toenjes, "3D Pumpkin Instructions," https://flic.kr/p/8M5jSC, 2010, The original image was freely adapted. Cited on Pages x and 5.

[11] Unknown, “A carved pumpkin sitting on top of a wooden porch. Carved pumpkin October Halloween.” https://picryl.com/media/carved-pumpkin-october-halloween-8be128, 2016. Cited on Pages x and 5.

[12] A. C. Antoulas, “Approximation of large-scale dynamical systems,” in *Advances in Design and Control*, 2005. Cited on Page 11.

[13] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019. Cited on Page 11.

[14] F. Chinesta, A. Huerta, G. Rozza, and K. E. Willcox, “Model Order Reduction: a survey,” 2016. Cited on Pages 11 and 21.

[15] M. Géradin and D. J. Rixen, *Mechanical Vibrations: Theory and Application to Structural Dynamics*. Wiley, 1994. Cited on Pages 11, 27, 53, 68, 93, 114, 124, and 132.

[16] D. C. Sorensen, *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*. Dordrecht: Springer Netherlands, 1997, pp. 119–165. Cited on Pages 12 and 72.

[17] W. C. Hurty, “Dynamic analysis of structural systems using component modes,” *AIAA Journal*, vol. 3, pp. 678–685, 1965. Cited on Page 13.

[18] R. R. Craig and M. Bampton, “Coupling of substructures for dynamic analyses.” *AIAA Journal*, vol. 6, pp. 1313–1319, 1968. Cited on Page 13.

[19] N. Bouhaddi and R. Fillod, “A method for selecting master DOF in dynamic substructuring using the Guyan condensation method,” *Computers & Structures*, vol. 45, no. 5, pp. 941–946, 1992. Cited on Page 13.

[20] U. L. Hetmaniuk and R. B. Lehoucq, “Multilevel methods for eigenspace computations in structural dynamics,” in *Domain Decomposition Methods in Science and Engineering XVI*, O. B. Widlund and D. E. Keyes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 103–113. Cited on Page 13.

[21] É. Balmès, “On the use of the energy norm for the reduction of structural dynamic models,” in *C. R. Acad. Sci. Paris*, vol. t. 323, Serie II b, 1996, pp. 255–260. Cited on Page 14.

[22] K. Kunisch and S. Volkwein, "Galerkin Proper Orthogonal Decomposition methods for parabolic problems," *Numerische Mathematik*, vol. 90, pp. 117–148, 2001. Cited on Page 14.

[23] I. Markovsky, *Low Rank Approximation: Algorithms, Implementation, Applications*. Springer, 2012. Cited on Page 14.

[24] F. Bamer and C. Bucher, "Application of the Proper Orthogonal Decomposition for linear and nonlinear structures under transient excitation," *Acta Mechanica*, vol. 223, 2012. Cited on Page 15.

[25] G. Kerschen, J.-C. Golinval, A. Vakakis, and L. Bergman, "The method of Proper Orthogonal Decomposition for dynamical characterization and order reduction of mechanical systems: An overview," *Nonlinear Dynamics*, vol. 41, pp. 147–169, 08 2005. Cited on Page 15.

[26] K. Lu, Y. Jin, Y. Chen, Y. Yang, L. Hou, Z. Zhang, Z. Li, and C. Fu, "Review for order reduction based on Proper Orthogonal Decomposition and outlooks of applications in mechanical systems," *Mechanical Systems and Signal Processing*, vol. 123, pp. 264–297, 2019. Cited on Page 15.

[27] K. Hoang, P. Kerfriden, and S. Bordas, "Space-time goal-oriented reduced basis approximation for linear wave equation," *arXiv: Computational Physics*, 05 2013. Cited on Page 15.

[28] K. C. Hoang, P. Kerfriden, B. Khoo, and S. Bordas, "An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems," *Numerical Methods for Partial Differential Equations*, vol. 31, 10 2014. Cited on Page 15.

[29] H. Fischer, J. Roth, T. Wick, L. Chamoin, and A. Fau, "MORe DWR: Space-time goal-oriented error control for incremental POD-based ROM for time-averaged goal functionals," *Journal of Computational Physics*, vol. 504, p. 112863, 2024. Cited on Pages 15 and 93.

[30] H. Fischer, J. Roth, L. Chamoin, A. Fau, M. Wheeler, and T. Wick, "Adaptive space-time model order reduction with dual-weighted residual (MORe DWR) error control for poroelasticity," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 11, no. 1, p. 9, Apr 2024. Cited on Page 15.

[31] S. Boyaval, C. Le Bris, T. Lelièvre, Y. Maday, N. C. Nguyen, and A. T. Patera, “Reduced basis techniques for stochastic problems,” *Archives of Computational methods in Engineering*, vol. 17, no. 4, pp. 435–454, 2010. Cited on Page 15.

[32] G. Rozza, D. B. P. Huynh, and A. T. Patera, “Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations,” *Archives of Computational Methods in Engineering*, vol. 15, no. 3, pp. 229–275, 2008. Cited on Page 15.

[33] É. Balmès, “Optimal Ritz vectors for component mode synthesis using the singular value decomposition,” *AIAA Journal*, vol. 34, pp. 1256–1260, 1996. Cited on Page 15.

[34] A. J. Keane and P. B. Nair, *Design in the Presence of Uncertainty.* John Wiley & Sons, Ltd, 2005, ch. 8, pp. 327–357. Cited on Pages 15 and 105.

[35] W. R. Hamilton, “On a general method in dynamics,” *Philosophical Transactions of the Royal Society of London*, vol. 124, pp. 247–308, 1834. Cited on Pages 16, 27, 31, and 36.

[36] ——, “Second essay on a general method in dynamics,” *Philosophical Transactions of the Royal Society of London*, vol. 125, pp. 95–144, 1835. Cited on Pages 16, 27, and 31.

[37] A. J. Lew, J. E. Marsden, M. Ortiz, and M. West, “An Overview of Variational Integrators,” in *Finite element methods: 1970’s and beyond*, 2004. Cited on Pages 17 and 80.

[38] D. Razafindralandy, A. Hamdouni, and M. Chhay, “A review of some geometric integrators,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 5, pp. 1–67, 2018. Cited on Pages 17 and 80.

[39] J. Simo and N. Tarnow, “The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics,” *Zeitschrift für angewandte Mathematik und Physik ZAMP*, vol. 43, pp. 757–792, 1992. Cited on Pages 17, 80, 104, and 137.

[40] L. Peng and K. Mohseni, “Symplectic model reduction of Hamiltonian systems,” *SIAM Journal on Scientific Computing*, vol. 38, p. A1–A27, 02 2016. Cited on Pages 18 and 80.

[41] B. M. Afkham and J. S. Hesthaven, “Structure preserving model reduction of parametric Hamiltonian systems,” *SIAM Journal on Scientific Computing*, vol. 39, no. 6, pp. A2616–A2644, 2017. Cited on Pages 18 and 80.

[42] P. Buchfink, S. Glas, and B. Haasdonk, “Optimal bases for symplectic model order reduction of canonizable linear Hamiltonian systems,” *IFAC-PapersOnLine*, vol. 55, no. 20, pp. 463–468, 2022, 10th Vienna International Conference on Mathematical Modelling MATHMOD 2022. Cited on Pages 18 and 80.

[43] A. Bunse-Gerstner, “Matrix factorizations for symplectic QR-like methods,” *Linear Algebra and its Applications*, vol. 83, pp. 49–77, 1986. Cited on Page 18.

[44] A. Salam and E. S. Al-Aidarous, “Equivalence between modified symplectic Gram-Schmidt and householder SR algorithms,” *BIT Numerical Mathematics*, vol. 54, pp. 283–302, 2014. Cited on Page 18.

[45] H. Sharma, Z. Wang, and B. Kramer, “Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical hamiltonian systems,” *Physica D: Nonlinear Phenomena*, vol. 431, p. 133122, 2022. Cited on Page 18.

[46] N. Shirafkan, P. Gosselet, F. Bamer, A. Oueslati, B. Markert, and G. de saxce, *Constructing the Hamiltonian from the Behaviour of a Dynamical System by Proper Symplectic Decomposition*, 07 2021, pp. 439–447. Cited on Page 18.

[47] P. Ladevèze, *PGD in linear and nonlinear Computational Solid Mechanics.* Vienna: Springer Vienna, 2014, pp. 91–152. Cited on Page 18.

[48] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. González, I. Alfaro, A. Ammar, and A. Huerta, “PGD-based computational vademecum for efficient design, optimization and control,” *Archives of Computational Methods in Engineering*, vol. 20, pp. 31–59, 2013. Cited on Pages 18, 20, and 22.

[49] L. Boucinha, A. Gravouil, and A. Ammar, “Space–time Proper Generalized Decompositions for the resolution of transient elastodynamic models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 255, pp. 67–88, 2013. Cited on Pages 21, 24, 27, 40, 41, and 65.

[50] L. Boucinha, “Réduction de modèle a priori par séparation de variables espace-temps : Application en dynamique transitoire,” Theses, INSA de Lyon, Nov. 2013, pp. 163–166. Cited on Pages 21, 22, 24, and 72.

[51] V. de Silva and L.-H. Lim, “Tensor rank and the ill-posedness of the best low-rank approximation problem,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008. Cited on Page 21.

[52] E. Musharbash, F. Nobile, and E. Vidlicková, "Symplectic dynamical low rank approximation of wave equations with random parameters," *BIT Numerical Mathematics*, pp. 1–49, 2020. Cited on Page 21.

[53] A. Nouy, "A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 23, pp. 1603–1626, 2010. Cited on Pages 21, 24, 27, and 47.

[54] S. Alameddin, A. Fau, D. Néron, P. Ladevèze, and U. Nackenhorst, "Toward optimality of Proper Generalised Decomposition bases," *Mathematical and computational applications*, vol. 24, no. 1, p. 30, Mar. 2019. Cited on Pages 21, 24, 27, and 47.

[55] L. Boucinha, A. Ammar, A. Gravouil, and A. Nouy, "Ideal minimal residual-based Proper Generalized Decomposition for non-symmetric multi-field models – Application to transient elastodynamics in space-time domain," *Computer Methods in Applied Mechanics and Engineering*, vol. 273, pp. 56–76, 2014. Cited on Pages 21, 24, and 70.

[56] A. Ammar, F. Chinesta, and A. Falcó, "On the Convergence of a Greedy Rank-One Update Algorithm for a Class of Linear Systems," *Archives of Computational Methods in Engineering*, vol. 17, pp. 473–486, 2010. Cited on Page 22.

[57] D. Goutaudier, L. Berthe, and F. Chinesta, "Proper Generalized Decomposition with time adaptive space separation for transient wave propagation problems in separable domains," *Computer Methods in Applied Mechanics and Engineering*, vol. 380, p. 113755, 2021. Cited on Page 22.

[58] A. Barbarulo, H. Riou, L. Kovalevsky, and P. Ladevèze, "PGD-VTCR: A reduced order model technique to solve medium frequency broad band problems on complex acoustical systems," *Strojniški Vestnik – Journal of Mechanical Engineering*, vol. 60, pp. 307–313, 2015. Cited on Page 22.

[59] P. De Brabander, O. Allix, P. Ladèveze, P. Hubert, and P. Thévenet, "On a wave-based reduced order model for transient effects computation including mid frequencies," *Computer Methods in Applied Mechanics and Engineering*, vol. 395, p. 114990, 2022. Cited on Page 22.

[60] C. Germoso, J. V. Aguado, A. Fraile, E. Alarcon, and F. Chinesta, "Efficient PGD-based dynamic calculation of non-linear soil behavior," *Comptes Rendus Mécanique*, vol. 344, no. 1, pp. 24–41, 2016. Cited on Page 22.

[61] M. H. Malik, D. Borzacchiello, J. V. Aguado, and F. Chinesta, "Advanced parametric space-frequency separated representations in structural dynamics: A harmonic–modal hybrid approach," *Comptes Rendus Mécanique*, vol. 346, no. 7, pp. 590–602, 2018. Cited on Page 22.

[62] G. Quaranta, C. Argerich Martin, R. Ibañez, J. L. Duval, E. Cueto, and F. Chinesta, "From linear to nonlinear PGD-based parametric structural dynamics," *Comptes Rendus Mécanique*, vol. 347, no. 5, pp. 445–454, 2019. Cited on Page 22.

[63] M. H. Malik, D. Borzacchiello, F. Chinesta, and P. Díez, "Inclusion of frequency-dependent parameters in power transmission lines simulation using harmonic analysis and Proper Generalized Decomposition," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 31, no. 5, p. e2331, 2018. Cited on Page 22.

[64] S. Rishmawi, S. Rodriguez, F. Chinesta, and F. P. Gosselin, "Harmonic-modal hybrid frequency approach for parameterized non-linear dynamics," *Computers & Structures*, vol. 301, p. 107461, 2024. Cited on Page 22.

[65] P. Díez, S. Zlotnik, A. García, and A. Huerta, "Encapsulated PGD algebraic toolbox operating with high-dimensional data," *Archives of Computational Methods in Engineering*, vol. 27, 11 2019. Cited on Page 22.

[66] F. Cavaliere, S. Zlotnik, R. Sevilla, X. Larráyoz, and P. Díez, "Nonintrusive reduced order model for parametric solutions of inertia relief problems," *International Journal for Numerical Methods in Engineering*, vol. 122, no. 16, pp. 4270–4291, 2021. Cited on Pages 22 and 134.

[67] F. Cavaliere, S. Zlotnik, R. Sevilla, X. Larrayoz, and P. Díez, "Nonintrusive parametric solutions in structural dynamics," *Computer Methods in Applied Mechanics and Engineering*, vol. 389, p. 114336, 2022. Cited on Pages 22 and 103.

[68] O. Zienkiewicz, R. Taylor, and J. Zhu, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*. Butterworth-Heinemann, 2013. Cited on Pages 22, 23, and 68.

[69] J. F. Semblat, "Rheological interpretation of Rayleigh damping," *Journal of Sound and Vibration*, vol. 206, no. 5, pp. 741–744, 1997. Cited on Pages 22, 23, and 68.

[70] A. Kareem and W.-J. Sun, "Dynamic response of structures with uncertain damping," *Engineering Structures*, vol. 12, no. 1, pp. 2–8, 1990. Cited on Pages 22, 23, and 68.

[71] F. Sánchez Iglesias and A. Fernández López, “Rayleigh damping parameters estimation using hammer impact tests,” *Mechanical Systems and Signal Processing*, vol. 135, p. 106391, 2020. Cited on Pages 23 and 105.

[72] S. Singh and M. Mastrogiuseppe, “Multibody modelling of tether and capture system for dynamic simulations of in-air capturing,” *Acta Astronautica*, vol. 218, pp. 59–69, 2024. Cited on Pages 23 and 105.

[73] C. A. Beattie and S. Gugercin, “Krylov-based model reduction of second-order systems with proportional damping,” *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2278–2283, 2005. Cited on Page 23.

[74] K. Meerbergen, “Fast frequency response computation for Rayleigh damping,” *International Journal for Numerical Methods in Engineering*, vol. 73, no. 1, pp. 96–106, 2008. Cited on Page 23.

[75] B. Wu, S. Yang, and Z. Li, “An algorithm for solving frequency responses of a system with rayleigh damping,” *Archive of Applied Mechanics*, vol. 86, 07 2016. Cited on Page 23.

[76] D. H. Hodges and R. R. Bless, “Weak Hamiltonian finite element method for optimal control problems,” *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 1, pp. 148–156, 1991. Cited on Pages 27, 36, and 39.

[77] J. L. Lagrange, *Méchanique Analitique.* Chez la Veuve Dessaint, 1788. Cited on Pages 27 and 31.

[78] T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Prentice-Hall, Inc., 1987. Cited on Page 27.

[79] G. M. Hulbert and T. J. Hughes, “Space-time finite element methods for second-order hyperbolic equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 84, no. 3, pp. 327–348, 1990. Cited on Pages 27, 53, and 65.

[80] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Computational Differential Equation.* Cambridge University Press, 01 1996. Cited on Pages 27 and 35.

[81] C. S. Jog and A. Nandy, “Conservation Properties of the Trapezoidal Rule in Linear Time Domain Analysis of Acoustics and Structures,” *Journal of Vibration and Acoustics*, vol. 137, no. 2, 2015, 021010. Cited on Pages 27 and 35.

[82] F. Iavernaro and D. Trigiante, "On some conservation properties of the trapezoidal method applied to Hamiltonian systems," in *ICNAAM 2005. International conference on numerical analysis and applied mathematics 2005. Official conference of the European Society of Computational Methods in Sciences and Engineering (ESCMSE), Rhodes, Greek, September 16–20, 2005.* Weinheim: Wiley-VCH, 2005, pp. 254–257. Cited on Pages 27 and 35.

[83] N. M. Newmark, "A Method of Computation for Structural Dynamics," *Journal of the Engineering Mechanics Division*, vol. 85, no. 3, pp. 67–94, 1959. Cited on Pages 27 and 53.

[84] M. Baruch and R. Riff, "Hamilton's principle, Hamilton's law - 6 to the $n$ power correct formulations," *AIAA Journal*, vol. 20, no. 5, pp. 687–692, 1982. Cited on Page 31.

[85] F. Chinesta, R. Keunings, and A. Leygue, *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer.* Springer, 2014. Cited on Pages 40, 41, 70, and 114.

[86] M. Billaud-Friess, A. Nouy, and O. Zahm, "A tensor approximation method based on ideal minimal residual formulations for the solution of high-dimensional problems," *Mathematical Modelling and Numerical Analysis*, vol. 48, pp. 1777–1806, 2014. Cited on Page 41.

[87] J.-Y. Cognard and P. Ladevèze, "The large time increment method applied to cyclic loadings," in *Creep in Structures*, M. Życzkowski, Ed. Springer Berlin Heidelberg, 1991, pp. 555–562. Cited on Page 41.

[88] P. Boisse, P. Bussy, and P. Ladevèze, "A new approach in non-linear mechanics: The large time increment method," *International Journal for Numerical Methods in Engineering*, vol. 29, no. 3, pp. 647–663, 1990. Cited on Page 47.

[89] G. Bonithon and A. Nouy, "A priori tensor approximations for the numerical solution of high dimensional problems: alternative definitions," in *28th GAMM-Seminar Leipzig on Analysis and Numerical Methods in Higher Dimensions*, Leipzig, Germany, Jan. 2012. Cited on Page 47.

[90] S. Rostami and R. Kamgar, "Insight to the Newmark implicit time integration method for solving the wave propagation problems," *Iranian Journal of Science and Technology - Transactions of Civil Engineering*, vol. 46, p. 679–697, 2021. Cited on Page 53.

[91] A. Ibrahimbegovic, H. C. Chen, E. L. Wilson, and R. L. Taylor, “Ritz method for dynamic analysis of large discrete linear systems with non-proportional damping,” *Earthquake Engineering & Structural Dynamics*, vol. 19, no. 6, pp. 877–889, 1990. Cited on Page 68.

[92] A. C. Aitken, “XII.—Further Numerical Studies in Algebraic Equations and Matrices,” *Proceedings of the Royal Society of Edinburgh*, vol. 51, p. 80–90, 1932. Cited on Page 72.

[93] P. Gosselet, C. Rey, and J. Pebrel, “Total and selective reuse of Krylov subspaces for the resolution of sequences of nonlinear structural problems,” *International Journal for Numerical Methods in Engineering*, vol. 94, no. 1, pp. 60–83, 2013. Cited on Pages 72 and 137.

[94] C. Vella and S. Prudhomme, “PGD reduced-order modeling for structural dynamics applications,” *Computer Methods in Applied Mechanics and Engineering*, vol. 402, p. 115736, 2022. Cited on Page 80.

[95] C. Cadiou, “Matplotlib label lines,” Dec. 2022. Cited on Pages xi and 96.

[96] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. Cited on Page 98.

[97] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. Cited on Pages 98 and 126.

[98] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, “Algorithm 887: CHOLMOD, Supernodal sparse cholesky factorization and update/downdate,” *ACM Trans. Math. Softw.*, vol. 35, no. 3, oct 2008. Cited on Pages 98, 113, and 126.

[99] K. Kergrene, L. Chamoin, M. Laforest, and S. Prudhomme, “On a goal-oriented version of the Proper Generalized Decomposition method,” *Journal of Scientific Computing*, vol. 81, p. 92–111, 2019. Cited on Pages 104 and 137.

[100] C. Vella, P. Gosselet, and S. Prudhomme, “An efficient PGD solver for structural dynamics applications,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 11, 07 2024. Cited on Pages 105, 107, 114, 117, 118, 119, and 132.

[101] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4. Cited on Pages 105 and 121.

[102] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73. Cited on Pages 105 and 121.

[103] L. J. V. Miranda, “PySwarms, a research-toolkit for Particle Swarm Optimization in Python,” *Journal of Open Source Software*, vol. 3, 2018. Cited on Page 126.

[104] L. Mandl, S. Goswami, L. Lambers, and T. Ricken, “Separable deeponet: Breaking the curse of dimensionality in physics-informed machine learning,” 2024. Cited on Page 137.

# APPENDIX A TIME OPERATORS

The computation of time integrals is required to evaluate the coefficients of the problem in space presented in Section 3.3.1, i.e. $k_t$, $c_t$, $d_t$, and $m_t$. Let $u = u(t)$ and $v = v(t)$ be two functions of time and assume they are sufficiently regular. We consider continuous, piecewise linear approximations of $u$ and $v$, which read in the case of $u$, and in a similar manner for $v$:

$$u(t) \simeq \left(1 - \frac{t - t^{i-1}}{h_t}\right) u\left(t^{i-1}\right) + \frac{t - t^{i-1}}{h_t} u\left(t^{i}\right), \quad t \in \left[t^{i-1}, t^{i}\right], \ i = 1, \dots, n_t,$$

with $h_t = t^i - t^{i-1}$. We can now define the vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{n_t}$ as:

$$\begin{aligned} \boldsymbol{u} &= \begin{bmatrix} u(t^0) & \dots & u(t^{n_t}) \end{bmatrix}^T, \\ \boldsymbol{v} &= \begin{bmatrix} v(t^0) & \dots & v(t^{n_t}) \end{bmatrix}^T. \end{aligned}$$

The time integrals are then approximated as:

$$\begin{aligned} \int_{\mathcal{I}} uv \; dt &\simeq \boldsymbol{u}^T A_t \boldsymbol{v}, \\ \int_{\mathcal{I}} \dot{u}v \; dt &\simeq \boldsymbol{u}^T C_t \boldsymbol{v}, \end{aligned}$$

with $A_t$ and $C_t$ the time operators such that:

$$A_t = \frac{h_t}{6} \begin{bmatrix} 2 & 1 & & & \\ & 4 & \ddots & [0] & \\ & & \ddots & \ddots & \\ & \text{sym.} & & 4 & 1 \\ & & & & 2 \end{bmatrix},$$

$$C_t = \frac{1}{2} \begin{bmatrix} \text{-1} & \text{-1} & & & \\ 1 & 0 & \ddots & [0] & \\ & \ddots & \ddots & \ddots & \\ & [0] & \ddots & 0 & \text{-1} \\ & & & 1 & 1 \end{bmatrix}.$$

# APPENDIX B DEFINITIONS OF THE PARAMETRIC PGD SUBPROBLEMS

The PGD formulation requires the computation of numerous coefficients. Their definitions are enumerated thereafter:

$$\begin{array}{llll}
k_x = \boldsymbol{\varphi}_q^T K \boldsymbol{\varphi}_q, & k_t = \displaystyle\int_{\mathcal{I}} \psi_q^2 \; dt, & k_\alpha = \displaystyle\int_{\mathcal{D}_\alpha} \xi_q^2 \; d\alpha, & k_\beta = \displaystyle\int_{\mathcal{D}_\beta} \zeta_q^2 \; d\beta, \\
m_x = \boldsymbol{\varphi}_p^T M^{-1} \boldsymbol{\varphi}_p, & m_t = \displaystyle\int_{\mathcal{I}} \psi_p^2 \; dt, & m_\alpha = \displaystyle\int_{\mathcal{D}_\alpha} \xi_p^2 \; d\alpha, & m_\beta = \displaystyle\int_{\mathcal{D}_\beta} \zeta_p^2 \; d\beta, \\
c_x = \boldsymbol{\varphi}_q^T \boldsymbol{\varphi}_p, & c_t = \displaystyle\int_{\mathcal{I}} \psi_q \dot{\psi}_p \; dt = \psi_q(T)\psi_p(T) - d_t, & c_\alpha = \displaystyle\int_{\mathcal{D}_\alpha} \xi_q \xi_p \; d\alpha, & c_\beta = \displaystyle\int_{\mathcal{D}_\beta} \zeta_q \zeta_p \; d\beta, \\
\xi_x = \boldsymbol{\varphi}_q^T M \boldsymbol{\varphi}_q, & \xi_t = \displaystyle\int_{\mathcal{I}} \psi_q \dot{\psi}_q \; dt = \frac{1}{2}\psi_q(T)^2, & \xi_\alpha = \bar{\alpha} \displaystyle\int_{\mathcal{D}_\alpha} \alpha \xi_q^2 \; d\alpha, & \xi_\beta = \bar{\beta} \displaystyle\int_{\mathcal{D}_\beta} \beta \zeta_q^2 \; d\beta.
\end{array}$$

With normalizations, the coefficients $k_x$, $m_x$, $k_\alpha$, $m_\alpha$, $k_\beta$ and $m_\beta$ are all set to unity.

**Problem in space.** The parameters $k_{qq}$, $m_{qq}$, $c_{qp}$, $c_{pq}$, and $m_{pp}$ are defined as follows:

$$\begin{aligned}
k_{qq} &= \xi_t \xi_\alpha k_\beta + k_t k_\alpha k_\beta, \\
m_{qq} &= \xi_t k_\alpha \xi_\beta, \\
c_{qp} &= c_t c_\alpha c_\beta, \\
c_{pq} &= d_t c_\alpha c_\beta, \\
m_{pp} &= -m_t m_\alpha m_\beta.
\end{aligned}$$

**Problem in time.** We assume that $\left(\boldsymbol{\varphi}_q, \xi_q, \zeta_q\right)$ and $\left(\boldsymbol{\varphi}_p, \xi_p, \zeta_p\right)$ are known and search for $\psi_q$ and $\psi_p$. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q^* \xi_q \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p^* \xi_p \zeta_p$:

$$\begin{aligned}
\int_{\mathcal{I}} \psi_q^{*T} \left[ c_\alpha c_\beta c_x \dot{\psi}_p + \xi_\alpha k_\beta k_x \dot{\psi}_q + k_\alpha \xi_\beta \xi_x \dot{\psi}_q + k_\alpha k_\beta k_x \psi_q \right] dt &= \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}_{m-1}^{q} \; dt d\mu, \quad \forall \psi_q^* \in C^0(\bar{\mathcal{I}}), \\
\int_{\mathcal{I}} \psi_p^{*T} \left[ c_\alpha c_\beta c_x \dot{\psi}_q - m_\alpha m_\beta m_x \psi_p \right] dt &= \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}_{m-1}^{p} \; dt d\mu, \quad \forall \psi_p^* \in C^0(\bar{\mathcal{I}}),
\end{aligned}$$

which simplifies to:

$$\begin{aligned}
c_\alpha c_\beta c_x \dot{\psi}_p + (\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) \dot{\psi}_q + k_\alpha k_\beta k_x \psi_q &= b_{\mathcal{T}}^{q}, \\
c_\alpha c_\beta c_x \dot{\psi}_q - m_\alpha m_\beta m_x \psi_p &= b_{\mathcal{T}}^{p},
\end{aligned}$$

with:

$$
\begin{aligned}
b_{\mathcal{T}}^{q} &= \int_{\mathcal{D}} \xi_q \zeta_q \boldsymbol{\varphi}_q^T \boldsymbol{r}_{m-1}^{q} \; d\mu, \\
b_{\mathcal{T}}^{p} &= \int_{\mathcal{D}} \xi_p \zeta_p \boldsymbol{\varphi}_p^T \boldsymbol{r}_{m-1}^{p} \; d\mu.
\end{aligned}
$$

Above equations are discretized using the Crank-Nicolson time-marching scheme, such that, given $\boldsymbol{\psi}^0$, one computes the $i^{\text{th}}$ iterate ($i > 0$) as:

$$
A_{\mathcal{T}} \boldsymbol{\psi}^i = B_{\mathcal{T}} \boldsymbol{\psi}^{i-1} + \boldsymbol{b}_{\mathcal{T}}^{i}, \qquad i = 1, \ldots, n_t,
$$

where:

$$
A_{\mathcal{T}} = \begin{bmatrix} h_t k_\alpha k_\beta k_x + 2(\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) & 2 c_\alpha c_\beta c_x \\ 2 c_\alpha c_\beta c_x & -h_t m_\alpha m_\beta m_x \end{bmatrix},
$$

$$
B_{\mathcal{T}} = \begin{bmatrix} -h_t k_\alpha k_\beta k_x + 2(\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) & 2 c_\alpha c_\beta c_x \\ 2 c_\alpha c_\beta c_x & h_t m_\alpha m_\beta m_x \end{bmatrix},
$$

$$
\boldsymbol{b}_{\mathcal{T}}^{i} = h_t \begin{bmatrix} b_{\mathcal{T}}^{q,i} + b_{\mathcal{T}}^{q,i-1} \\ b_{\mathcal{T}}^{p,i} + b_{\mathcal{T}}^{q,i-1} \end{bmatrix}.
$$

**Problem in damping parameter $\alpha$.** We assume that $\left(\boldsymbol{\varphi}_q, \psi_q, \zeta_q\right)$ and $\left(\boldsymbol{\varphi}_p, \psi_p, \zeta_p\right)$ are known and search for $\xi_q$ and $\xi_p$. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q \xi_q^* \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p \xi_p^* \zeta_p$:

$$
\begin{aligned}
\int_{\mathcal{D}_\alpha} \xi_q^* \left( c_t c_\beta c_x \xi_p + \xi_t k_\beta k_x \alpha \bar{\alpha} \xi_q + \xi_t \xi_\beta \xi_x f_q + k_t k_\beta k_x \xi_q \right) d\alpha &= \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}_{m-1}^{q} \; dt d\mu, \quad \forall \xi_q^* \in C^0(\mathcal{D}_\alpha), \\
\int_{\mathcal{D}_\alpha} \xi_p^* \left( d_t c_\beta c_x \xi_q - m_t m_\beta m_x \xi_p \right) d\alpha &= \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}_{m-1}^{p} \; dt d\mu, \quad \forall \xi_q^* \in C^0(\mathcal{D}_\alpha),
\end{aligned}
$$

which simplifies to:

$$
\begin{aligned}
c_t c_\beta c_x \xi_p + (\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha \bar{\alpha}) \xi_q &= b_\xi^q, \\
d_t c_\beta c_x \xi_q - m_t m_\beta m_x \xi_p &= b_\xi^p,
\end{aligned}
$$

with:

$$
\begin{aligned}
b_\xi^q &= \int_{\mathcal{D}_\beta} \int_{\mathcal{I}} \psi_q \zeta_q \boldsymbol{\varphi}_q^T \boldsymbol{r}_{m-1}^{q} \; dt d\beta, \\
b_\xi^p &= \int_{\mathcal{D}_\beta} \int_{\mathcal{I}} \psi_p \zeta_p \boldsymbol{\varphi}_p^T \boldsymbol{r}_{m-1}^{p} \; dt d\beta.
\end{aligned}
$$

The solutions read, with dependencies to $\alpha$ explicitly written:

$$\xi_q(\alpha) = \frac{c_t c_\beta c_x b_\xi^p(\alpha) + m_t m_\beta m_x b_\xi^q(\alpha)}{(\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) m_t m_\beta m_x + c_t d_t (c_\beta c_x)^2},$$
$$\xi_p(\alpha) = \frac{d_t c_\beta c_x b_\xi^q(\alpha) - (\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) b_\xi^p(\alpha)}{(\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) m_t m_\beta m_x + c_t d_t (c_\beta c_x)^2}.$$

**Problem in parameter $\beta$.** We assume that $\left(\boldsymbol{\varphi}_q, \psi_q, \xi_q\right)$ and $\left(\boldsymbol{\varphi}_p, \psi_p, \xi_p\right)$ are known and search for $\zeta_q$ and $\zeta_p$. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q \xi_q \zeta_q^*$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p \xi_p \zeta_p^*$:

$$\int_{\mathcal{D}_\beta} \zeta_q^* \left[c_t c_\alpha c_x \zeta_p + \xi_t \xi_\alpha k_x \zeta_q + \xi_t k_\alpha \xi_x \beta\bar{\beta} \zeta_q + k_t k_\alpha k_x \zeta_q\right] d\beta = \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}_{m-1}^q \, dt d\mu, \quad \forall \zeta_q^* \in C^0(\mathcal{D}_\beta),$$
$$\int_{\mathcal{D}_\beta} \zeta_p^* \left[d_t c_\alpha c_x \zeta_q - m_t m_\alpha m_x \zeta_p\right] d\beta = \int_{\mathcal{D}} \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}_{m-1}^p \, dt d\mu, \quad \forall \zeta_q^* \in C^0(\mathcal{D}_\beta),$$

which simplifies to:

$$c_t c_\alpha c_x \zeta_p + (\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) \zeta_q = b_\zeta^q,$$
$$d_t c_\alpha c_x \zeta_q - m_t m_\alpha m_x \zeta_p = b_\zeta^p,$$

with:

$$b_\zeta^q = \int_{\mathcal{D}_\alpha} \int_{\mathcal{I}} \psi_q \xi_q \boldsymbol{\varphi}_q^T \boldsymbol{r}_{m-1}^q \, dt d\alpha,$$
$$b_\zeta^p = \int_{\mathcal{D}_\alpha} \int_{\mathcal{I}} \psi_p \xi_p \boldsymbol{\varphi}_p^T \boldsymbol{r}_{m-1}^p \, dt d\alpha.$$

The solutions read, with dependencies to $\beta$ explicitly written:

$$\zeta_q(\beta) = \frac{c_t c_\alpha c_x b_\zeta^p(\beta) + m_t m_\alpha m_x b_\zeta^q(\beta)}{(\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) m_t m_\alpha m_x + c_t d_t (c_\alpha c_x)^2},$$
$$\zeta_p(\beta) = \frac{d_t c_\alpha c_x b_\zeta^q(\beta) - (\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) b_\zeta^p(\beta)}{(\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) m_t m_\alpha m_x + c_t d_t (c_\alpha c_x)^2}.$$