

Nouvelles représentations séparées pour des applications industrielles complexes: espace, temps et paramètres Angelo Pasquale

► To cite this version:

Angelo Pasquale. Nouvelles représentations séparées pour des applications industrielles complexes : espace, temps et paramètres. Mécanique des matériaux [physics.class-ph]. HESAM Université, 2024. Français. NNT : 2024HESAE009 . tel-04526130

HAL Id: tel-04526130 https://pastel.hal.science/tel-04526130

Submitted on 29 Mar 2024 $\,$

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.







ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR

Laboratoire Angevin de Mécanique, Procédés et innovAtion (LAMPA) – Campus d'Angers Laboratoire Procédés et Ingénierie en Mécanique et Matériaux (PIMM) – Campus de Paris

THÈSE

présentée par

Angelo PASQUALE

soutenue le 26 janvier 2024

pour obtenir le grade de

Docteur d'HESAM Université

préparée à

l'École Nationale Supérieure d'Arts et Métiers Spécialité : Mécanique-matériaux

Nouvelles représentations séparées pour des applications industrielles complexes : espace, temps et paramètres

Thèse dirigée par

Prof. Amine Ammar

et co-encadrée par

Dr. Mustapha ZIANE

soutenue devant le **jury** composé de :

м.	Fodil Meraghni	PR, Arts et Métiers	Président
м.	Florian DE VUYST	PR, UTC	Rapporteur
м.	David González	PR, Universidad de Zaragoza	Rapporteur
м.	Amine Ammar	PR, Arts et Métiers	Directeur
м.	Mustapha ZIANE	DR, ESI Group	Co-encadrant
Mme.	Marianne BERINGHIER	MCF, ISAE–ENSMA	Examinatrice
м.	Nawfal Blal	MCF, INSA Lyon	Invité
м.	Francisco CHINESTA	PR, Arts et Métiers	Invité
м.	Christian Ellersdorfer	PR, Technische Universität Graz	Invité

T H È S E

Le verità che contano, i grandi principi, alla fine, restano due o tre. Sono quelli che ti ha insegnato tua madre da bambino. — Enzo BIAGI

REMERCIEMENTS

Je commence par remercier infiniment trois personnes clés pour le déroulement de mes travaux de thèse : Mustapha Ziane, Amine Ammar et Paco Chinesta. Je suis extrêmement reconnaissant envers mon encadrant industriel, Mustapha, pour sa bienveillance, sa transparence et sa disponibilité, y compris le weekend, pour approfondir chaque détail, qu'il soit théorique, implémentatif ou applicatif, des sujets que nous avons traités ensemble. Je tiens à exprimer ma profonde gratitude envers Amine, mon directeur de thèse, pour la confiance qu'il m'a accordée dès mon premier stage avec lui. Malgré nos interactions à distance, parfois à travers le monde, sa réactivité et sa disponibilité ont été des atouts majeurs qui nous ont permis de progresser efficacement. Je souhaite adresser mes remerciements les plus sincères à Paco, *the boss*, pour son énergie débordante, sa passion contagieuse, sa bonne humeur constante et son amitié précieuse. "Models are models, reality is reality", mais toi tu resteras toujours a *real model*, merci *boss* !

Je tiens à remercier tous les membres du jury pour avoir consacré leur temps à examiner et évaluer mes travaux. Je suis reconnaissant de leur intérêt pour les recherches de l'équipe, de leurs retours détaillés et constructifs sur le manuscrit, ainsi que de nos échanges enrichissants.

Un grand merci aux équipes d'AMValor pour le cadre de travail optimal qu'elles ont créé. Je suis reconnaissant envers ESI Group pour le financement de cette thèse via la chaire de recherche CREATE-ID aux Arts et Métiers, ainsi que pour les défis stimulants liés à l'industrie.

Je tiens à exprimer ma gratitude envers Dominique Baillargeat et les équipes de CNRS@CREATE, pour leurs accueil à Singapour, lors de ma participation au projet DesCartes. Tous ces voyages ont été extrêmement enrichissants à la fois sur le plan personnel et professionnel.

Un grand merci à tous mes collègues doctorants, postdocs, enseignants-chercheurs et ingénieurs car chacun d'entre eux a contribué à ma formation de différentes manières.

Merci Khanh et Sebastian pour les travaux que nous avons menés ensemble dans la multi-échelle temporelle, et pour croire autant que moi au potentiel de ces travaux pour de nombreuses applications.

Merci à Chady pour sa disponibilité constante, sa sympathie et ses précieuses contributions sur le plan/hors-plan, ainsi que pour son *marteau* en alliage de réseaux de neurones imbattables.

Thank you very much, Javad, for tackling several challenges with me in numerous research projects, especially those concerning the applications of the NURBS-PGD method or modular PGD.

Dankeschön an Alexander für sein Engagement in unserer Zusammenarbeit bei Safe-LIB, das meiner Dissertation einen echten Mehrwert bringt. Ich habe deine herzliche Aufnahme in Österreich und deine perfekte Abstimmung mit mir bei der Entwicklung neuer F&E-Projekte wirklich geschätzt.

Merci à Sergio d'avoir toujours fait confiance à mes modestes compétences CFD et LAT_EX . Merci aussi de nous avoir transportés optimalement même de Kyoto à Osaka, et de rêver de la commande pip install iP••p.

Merci (et désolé au passage) à Nicolas pour les milles commandes de PC !

Merci, Simon, pour ta super vibe depuis le début, au travail comme en dehors. Merci pour nos soirées de *ouf*, et même pour avoir toujours *kiffé* et imité mon accent. Merci aussi de m'avoir rendu une petite star auprès de tes potes. T'es trop *chanmé*, mec !

Merci à Victorigno, ma première rencontre au PIMM, qui est vite devenu un excellent compagnon de voyage, avec une valise Air France remplie de plaisanteries et connaissances. Encore une fois je te remercie pour cette synergie de *cartésien* et *pragmatique*, qui m'apporte bien plus de ce que t'imagines !

Caro Danielino, ho prababilmente detto più volte grazie (e scusa) a te che a $\iota \chi \theta \dot{\upsilon} \varsigma$! Ancora una volta, ti ringrazio per starmi dietro dal primo giorno in cui mi chiedesti gli appunti di Salsa in Nave. Grazie per quella marcia in più che mi dai, per la tua minuziosità, semplicità e umanità. Grazie di sopportare, contenere, e regolare i miei mille sbalzi d'umore e le mie duemila fisse.

Grazie agli amici di \mathfrak{GLG} perché alla fine questo dottorato è stata la continuazione naturale del percorso avviato assieme, quindi c'è ancora del vostro. Siamo tutti sparsi per il mondo, ma allo stesso tempo tutti riuniti! Grazie per essere stati sempre pronti a venire a trovarci qui in Francia, che sia per un doppio carpiato in piscina dalla Leti (alle sue fenomenali cerimonie di apertura dell'estate – grazie ancora Tiziapolina!), per dormire su un vecchio materassino nel salotto di Villejuif, oppure per una raclette dalla Lux. Tra l'altro, grazie Lux per avermi accolto dal primissimo giorno sul tuo divano (probabilmente c'è della posta che arriva ancora da te), per avermi fatto incontrare i primi amici parigini e scoprire le prime serate *Driiing*, di cui sono presto diventato PR. Grazie mille al grande *FraNGuozz*, perché in questi tre anni a separarci non c'è stata manco una porta, grazie per le infinite chiacchierate, per tutte le avventure pazzerelle, i nostri sbrocchi e le nostre risate.

Merci Jimmy de m'accompagner dans milles aventures depuis mon premier jour en France, et d'être un vrai *fra* sans filtres (gardes-en pour les clopes, par contre) !

Merci P.A. pour tes lettres profondes, pour le sacré personnage que tu es et toute la détermination que tu transmets.

Merci à toutes les belles rencontres et aux amis de Paris, car avec vous je n'ai pas vu les weekends passer (d'ailleurs, des fois, même pas la paie arriver !). Je pense notamment à tous les moments intenses, drôles ou sérieux, joyeux ou tristes, vécus avec *les fêtards Parisch'* et *la team 77 One Life*.

Grazie agli amori di Angela e Roby, per esservi rivelate una vera scoperta in questi ultimi anni di ricerca. Farvi da guida per la città, sconvolgervi e stregarvi, tutto una grande *kiffance*. Nonostante non vi abbia fatto vedere altro, essere stato il primo a portarvi a veder le stelle dal bois de Vincennes è davvero un'immensa soddisfazione!

Grazie a tutti gli amici di giù, perché più gli anni passano e le esperienze si accumulano, più realizzo il lusso e la fortuna che ho ad avere questi legami che si autoalimentano ed intensificano nonostante la distanza.

Grazie a tutta la mia famiglia. Questo dottorato è, come ogni altro mio traguardo, anche vostro. Grazie a tutte le zie e gli zii, le cugine e i cugini, le nonne e i nonni, per il vostro prezioso sostegno e i vostri valori, che mi accompagnano, da sempre, in ogni parte del mondo.

Grazie mamma per accompagnare sempre le mie orme sulla sabbia e, soprattutto, per quelle volte in cui c'è una sola orma! Grazie papà perché hai sempre desiderato il meglio per noi e, anche nei momenti più cupi, sei riuscito a darcelo; grazie per continuare ad ascoltarci, ad interessarti, ma, soprattutto, a batterti assieme a noi nelle tante sfide che affrontiamo! Grazie *Pippi* perché da te non smetto mai di imparare e perché tu non smetti mai di far di me la tua prima preoccupazione; grazie perché, sin da piccolo, mi hai insegnato e dimostrato che se puoi sognarlo puoi farlo, che in due si può lottare come dei giganti contro ogni dolore, e che le tempeste fanno sì che gli alberi mettano radici più forti!

Et, enfin, je remercie de tout mon cœur Matilde, pour son soutien inconditionnel, qui va bien au-delà de tout aspect scientifique de cette thèse. Merci pour ton énergie scintillante, ton affection chaleureuse, ta joie communicative, et ta *conchiglia* de réconfort quand tout est chaos à côté, quand rien n'a de sens, et rien ne va ! Grazie di cuore, *Coulmi*, perché mi fai volare sempre più in alto!

NOVEL SEPARATED REPRESENTATIONS FOR CHALLENGING INDUSTRIAL APPLICATIONS: SPACE, TIME AND PARAMETERS

ABSTRACT

Every transient problem in continuum mechanics is characterized by three variables: space, time and parameters. The space defines the physical domain, enabling the definition of diverse systems. The time captures dynamic processes, allowing for transient behavior analysis. The parameters control system and modeling characteristics. Together, these elements drive the accuracy and relevance of computational science, making them essential for understanding and predicting real-world phenomena.

The complexity arising from managing space, time, and parameters in numerical simulations can be particularly challenging in some scenarios. This is the case, for instance, when dealing with thin structures, small time steps combined with long time intervals, and a high number of parameters over large domains.

The numerical simulation of three-dimensional models in thin geometries presents important challenges since maintaining the mesh granularity proportional to the thickness dimension requires an impractical number of elements for the entire structure. This issue is currently encountered in automotive industry when considering vehicle crash simulations, where most of the components are thin structures.

When time multiscale behaviours occur, standard discretization techniques are constraint to mesh up to the finest scale to predict accurately the response of the system. This results in a prohibitive computational when the phenomena must be observed over a long duration. This occurs, for instance, in material science when dealing with fatigue damage assessments and cyclic visco-elasto-plastic fatigue problems.

A large number of parameters increases the dimensionality of the parameter space exponentially, making its exploration computationally intensive. The data generated from numerous simulations can be difficult to manage and advanced meta-modeling techniques are required. This typically happens in optimal design problems of multicomponent parametric structures.

To address these challenges, it is essential to strike a balance between accuracy and computational efficiency, requiring ad-hoc advanced developments. In this thesis the three challenges are separately addressed via novel separation-based techniques.

NOUVELLES REPRÉSENTATIONS SÉPARÉES POUR DES APPLICATIONS INDUSTRIELLES COMPLEXES : ESPACE, TEMPS ET PARAMÈTRES

Résumé

Chaque problème transitoire en mécanique des milieux continus est caractérisé par trois variables : l'espace, le temps et les paramètres. L'espace définit le domaine physique, permettant la description de divers systèmes. Le temps capture les processus dynamiques, autorisant l'analyse des comportements transitoires. Les paramètres influent sur les caractéristiques des systèmes et de la modélisation. Ensemble, ces éléments influent sur l'exactitude et la pertinence de la science computationnelle, les rendant essentiels pour la compréhension et la prédiction des phénomènes réels.

La gestion de l'espace, du temps et des paramètres dans les simulations numériques est particulièrement impactée dans certains scenarios. Cela est le cas, par exemple, lorsqu'il s'agit de structures minces, de petits pas de temps combinés à de longs intervalles, et d'un grand nombre de paramètres sur de vastes domaines.

La simulation numérique de modèles tridimensionnels dans des géométries minces présente d'importants défis, car maintenir la granularité du maillage proportionnelle à la dimension de l'épaisseur nécessite un nombre impraticable d'éléments pour toute la structure. Cela survient souvent dans l'industrie automobile lors de simulations de crash, où la plupart des composants sont des structures minces.

Lorsque des comportements multiscales dans le temps surviennent, les techniques de discrétisation standard sont contraintes de mailler jusqu'à l'échelle la plus fine pour prédire avec précision la réponse du système. Cela entraîne un coût computationnel prohibitif lorsque les phénomènes sont observés sur une longue durée, comme c'est le cas en science des matériaux lors de l'évaluation des endommagements par fatigue.

Un grand nombre de paramètres augmente de manière exponentielle la dimension de l'espace paramètrique, limitant son exploration. Les données générées par de nombreuses simulations peuvent être difficiles à gérer et des techniques avancées de métamodélisation sont nécessaires. Cela se produit généralement dans des problèmes de conception optimale de structures paramétriques à plusieurs composants.

Pour relever ces défis, il est essentiel de trouver un équilibre entre la précision et l'efficacité computationnelle, nécessitant des développements avancés. Dans cette thèse, les trois défis sont abordés via des nouvelles techniques basées sur les représentations séparéés.

ix

Résumé etendu

CONTEXTE ET MOTIVATIONS

L'espace, le temps et les paramètres sont les trois variables fondamentales décrivant tout système physique. L'espace se réfère à la région dans laquelle les objets physiques et les phénomènes sont observés. Le temps régit l'évolution des différents processus. Les paramètres sont essentiels pour construire des modèles mathématiques des phénomènes physiques, tenant compte des propriétés spécifiques et des caractéristiques des systèmes analysés. Par conséquent, ces variables jouent un rôle clé dans l'ingénierie basée sur la simulation [1–3], où des modèles fiables nécessitent des descriptions précises de la région spatiale, de l'évolution temporelle et des analyses de sensibilité aux paramètres adéquates.

Dans les simulations numériques [4,5], le domaine spatial est généralement discrétisé en éléments (c'est-à-dire, des éléments solides lors de la considération d'un espace tridimensionnel) ou en points de grille, définissant ce qu'on appelle le maillage computationnel. De même, le domaine temporel est discrétisé en petits pas de temps. Les équations et modèles gouvernants sont ensuite transférés dans leurs équivalents discrets, adaptés au calcul numérique via la méthode des éléments finis (MEF) et ses variantes. Le domaine paramétrique est exploré en résolvant les équations pour différentes valeurs des paramètres et en observant les changements résultants dans la réponse du système. Il est bien connu que la taille du maillage affecte la précision et la stabilité de la simulation, mais aussi le temps de calcul, car des maillages plus fins entraînent des coûts computationnels plus élevés. Cela se produit également dans les analyses paramétriques, où un plus grand nombre de paramètres ou des plages de paramètres plus vastes nécessitent un temps de calcul plus long pour explorer le domaine paramétrique.

En réalité, malgré les progrès réalisés en analyse numérique et en informatique haute performance, de nombreux problèmes du monde réel et des systèmes à grande échelle peuvent entraîner des simulations prohibitives en termes de coût. La réduction d'ordre de modèle (ROM) [6,7] est la branche de la science computationnelle qui aborde cette question, cherchant le juste équilibre entre la qualité de la simulation et l'efficacité. Cette thèse vise à enrichir les techniques actuelles de ROM sur trois niveaux, à savoir 1. l'espace, 2. le temps et 3. les paramètres, motivée par trois scénarios difficiles différents rencontrés de nos jours dans l'ingénierie basée sur la simulation.

En ce qui concerne l'espace, le travail se concentre sur la nécessité d'un cadre de simulation efficace, valide des géométries minces aux géométries épaisses en mécanique des structures. En ce qui concerne le temps, le défi consiste à combiner une très longue durée des phénomènes avec un très petit pas de temps, pour des problèmes complexes et non linéaires en mécanique des solides. Enfin, en ce qui concerne les paramètres, le travail se concentre sur des problèmes de conception multi-composants impliquant un grand nombre de paramètres de modélisation.

Chaque défi est donc motivé par une dimension caractéristique dans le système. Pour l'espace, cette dimension est l'épaisseur de la structure. Pour le temps, il s'agit du nombre de cycles de charge ou de la durée de la simulation. Pour les paramètres, cela correspond au nombre de paramètres.

Les motivations de ces trois axes sont exposées plus en détail ci-dessous, en omettant un aperçu approfondi et une description de la recherche et de la littérature les plus récentes sur les sujets, qui seront fournies par la suite dans chaque chapitre associé.

ESPACE

La résolution de modèles 3D dans des géométries dégénérées, où une dimension caractéristique est inférieure aux autres, n'est pas trivial. C'est le cas des plaques et des coques, où l'épaisseur (souvent appelée dimension hors plan) est plus petite par rapport à ses autres dimensions (appelées dimensions dans le plan) [8]. Pour citer quelques exemples de telles structures, en ingénierie, elles sont rencontrées dans les composants et les corps d'automobiles, d'aéronefs ou de navires, dans les dispositifs micro-électromécaniques ou les batteries électriques, dans les pipelines et les réservoirs. En architecture, elles font souvent partie de ponts, de bâtiments et de dômes. Dans la nature, on peut simplement imaginer des œufs, des mollusques, des squelettes, des membranes biologiques ou la peau humaine.

Lors de la discrétisation d'un domaine dégénéré de ce type, le rapport d'aspect des éléments de maillage (c'est-à-dire le rapport entre le côté le plus long de l'élément et le côté le plus court) peut devenir important. Cependant, des éléments avec de mauvais rapports d'aspect peuvent conduire à une instabilité numérique, des imprécisions et des problèmes de convergence. Pour éviter cela, la granularité du maillage doit s'ajuster à la dimension de l'épaisseur, nécessitant un nombre d'éléments impraticable pour mailler l'ensemble de la structure.

Cette propriété limite l'applicabilité de la méthode des éléments finis (MEF) traditionnelle basée sur des éléments solides 3D, conduisant à la nécessité de développer des éléments spécialisés, appelés éléments coque 2D [9,10]. Ceux-ci reposent sur la théorie des coques, qui fait certaines hypothèses sur le comportement à travers l'épaisseur. De la même manière, les structures longues et élancées sont généralement discrétisées à l'aide des éléments poutre 1D, qui reposent sur la théorie de la poutre.

Malgré leur efficacité dans de nombreux scénarios, les éléments de coque présentent également plusieurs inconvénients : (a) ils sont principalement conçus pour analyser le comportement dans le plan, et peuvent ne pas bien fonctionner dans des situations où les déformations hors plan ou les effets de flambage sont significatifs ; (b) ils sont généralement utilisés pour des analyses linéaires, et peuvent ne pas être adaptés aux analyses non linéaires, telles que l'élasticité non linéaire due aux non linéarités matérielles ou géométriques, les comportements élastoplastiques ou les comportements complexes de multiphysique ; (c) la théorie de base des coques est développée pour des structures de coques minces et les résultats peuvent être compromis lors du passage de coques minces à modérément épaisses ou épaisses, où des formulations d'ordre supérieur et des éléments coque épais spécifiques sont nécessaires (généralement, le choix du bon type d'élément n'est pas évident).

Cependant, les éléments coque classiques restent encore le choix le plus courant dans l'industrie, en raison de leur efficacité. Pour donner une idée, dans un véhicule complet, on peut compter environ 10^2 éléments poutre, 10^6 éléments coque et 10^4 éléments solides. Une méthode actuellement utilisée pour contourner les problèmes mentionnés ci-dessus consiste à effectuer, pendant la simulation, un remaillage coque vers solide pour certaines parties spécifiques, lorsque les résultats commencent à paraître douteux et lorsque des effets physiques plus complexes doivent être capturés. Cela nécessite une extrusion de la pièce suivie d'un remaillage impliquant un grand nombre d'éléments solides, ce qui augmente considérablement le temps de calcul. De plus, cette procédure entraîne également d'autres difficultés telles que le couplage de la partie extrudée

avec les éléments poutre 1D ou les éléments coque 2D environnants, ce qui n'est généralement pas trivial.

Si l'on considère la structure de la caisse en blanc (CEB), presque toutes les pièces sont des structures minces, comme le montre la figure 1, où trois d'entre elles sont mises en évidence en orange.



FIGURE 1: Exemples de structures minces dans la caisse en blanc (avec l'aimable autorisation du groupe ESI).

Dans ce contexte, l'objectif principal de la thèse est de suggérer et d'explorer des voies alternatives pour un cadre de simulation efficace, valide de structures coques minces à épaisses.

TEMPS

Malgré la disponibilité de plates-formes informatiques haute performance, la solution numérique de problèmes complexes, dépendant du temps et non linéaires, peut encore de nos jours représenter un défi ardu. Cette problématique est encore accentuée dans le cas de comportements multi-échelles, où des effets se produisent à différentes échelles spatiales et temporelles. Dans de tels contextes, le traitement des différentes échelles du problème peut être obligatoire pour définir des modèles fiables. Ces défis sont rencontrés en mécanique des solides numérique (CSM) [11], lorsqu'il s'agit de métamatériaux, de fabrication additive, de formage de tôles, de viscoplasticité cyclique ou de dynamique avec des chargements impliquant plusieurs échelles de temps caractéristiques.

En tant que défi majeur, cette situation devient extrêmement délicate lorsque les phénomènes doivent être simulés sur des intervalles de temps très longs et, en même temps, la réponse du système doit englober les différentes échelles de temps présentes dans le modèle. Les techniques de discrétisation standard sont contraintes de mailler jusqu'à l'échelle la plus fine pour prédire avec précision la réponse du système, ce qui entraîne une augmentation dramatique des coûts informatiques et, dans certains cas, devient informatiquement irréalisable. De plus, de tels scénarios entraînent d'autres problèmes tels que l'accumulation d'erreurs d'arrondi numériques, la saturation des ressources de stockage ou la nécessité de pas de temps adaptatifs.

Prenons le cas de simulations fines de fatigue cyclique, où les dommages de fatigue s'accumulent progressivement à chaque cycle. Pour donner un ordre de grandeur, le nombre de cycles jusqu'à la défaillance en fatigue se situe entre 10^4-10^5 dans le cas de la fatigue à faible cycle (LCF), 10^5-10^7 pour la fatigue à grand cycle (HCF), et plus de 10^7 pour la fatigue à très grand cycle (VHCF). Par exemple, la plupart des

applications portantes dans les véhicules, les pièces de moteurs, sont soumises à 10^8 cycles, tandis que les composants ferroviaires, les ponts et les roues sont soumis à 10^9 cycles au cours de leur durée de vie [12,13]. De plus, pour une simulation à fine échelle, 10^3 pas de temps pourraient être nécessaires sur un seul cycle. Ces chiffres montrent directement que simuler chaque cycle individuel est irréalisable.

Au lieu de simuler tous les cycles individuels, d'autres techniques sont utilisées, telles que des approches statistiques, des modèles d'accumulation de dommages, des techniques de simplification telles que le comptage rainflow ou la règle de Miner [14–16].

Dans ce contexte, la thèse vise à proposer une nouvelle procédure de multi-échelle temporelle et à enquêter sur son efficacité dans des problèmes non linéaires.

PARAMÈTRES

La plupart des applications et des processus en ingénierie nécessitent des études paramétriques, où la dépendance paramétrique peut être de nature à la fois physique (par exemple, les coefficients du modèle, les termes sources) et géométrique (par exemple, la forme du domaine). Cela se produit, par exemple, dans l'optimisation, la quantification et la propagation de l'incertitude, l'identification inverse ou le contrôle basé sur la simulation. Dans tous ces scénarios nécessitant de nombreuses requêtes, des évaluations quasi temps réel de la réponse du système sont nécessaires. Malgré les progrès réalisés dans la réduction d'ordre du modèle (MOR) et l'apprentissage automatique (ML) [6,7] pour construire des approximations paramétriques, la tâche reste difficile lorsqu'il s'agit de grands domaines, de systèmes à grande échelle, combinés à un grand nombre de paramètres.

En fait, à mesure que le nombre de paramètres augmente, la dimensionnalité du problème augmente de manière exponentielle (malédiction de la dimension) et un grand nombre de simulations, chacune impliquant différentes combinaisons de paramètres, sont nécessaires pour explorer l'espace des paramètres, sollicitant les ressources informatiques, la mémoire et le stockage. De plus, dans un espace de paramètres de grande dimension, les paramètres peuvent être interdépendants, ce qui rend difficile l'identification des combinaisons les plus pertinentes à étudier. Cela peut conduire à l'exploration d'ensembles de paramètres redondants ou non informatifs. D'autres limitations concernent l'identification des ensembles de paramètres optimaux, l'étude de la sensibilité des paramètres, mais aussi l'interprétabilité et la validation du modèle lui-même.

Ce problème se rencontre dans les structures paramétriques multi-composants, où divers composants ou pièces interagissent et leurs comportements dépendent de différents paramètres. Pour donner une idée des difficultés, prenons à nouveau un problème industriel en ingénierie automobile. La construction d'une solution de crash paramétrique d'une structure composée de 200 pièces, chacune impliquant un matériau paramétrique décrit par 3 paramètres et l'épaisseur de la pièce. Ainsi, le nombre de paramètres devient exorbitant, atteignant 800. Dans les technologies de régression les plus avancées, le nombre de données (solutions de problèmes haute résolution) évolue avec le nombre de paramètres. Dans le cas présent, même en utilisant les techniques de régression les plus avancées, plus de 800 solutions haute-fidélité semblent nécessaires, mais compte tenu du fait que chacune de ces solutions implique une journée de calcul, le coût computationnel représente 3 années de calculs. La figure 2 illustre le défi, mettant en évidence quelques composants paramétriques possibles dans le pilier B d'un véhicule.



FIGURE 2: Exemple d'une structure multiparamétrique dans l'industrie automobile (avec l'aimable autorisation du groupe Hyundai Motor).

Représentations séparées

Considérons un problème différentiel générique défini sur la région spatiale $\Omega \subset \mathbb{R}^n$ (avec n = 1, 2, 3), évoluant dans l'intervalle de temps $I \subset \mathbb{R}$ et incorporant des paramètres résidant dans l'espace paramétrique $\Theta \subset \mathbb{R}^d$ (avec $d \in \mathbb{N}$). Soit $\boldsymbol{x} = (x, y, z) \in \Omega$ le vecteur des coordonnées spatiales, $t \in I$ la coordonnée temporelle et $\boldsymbol{p} = (p_1, \ldots, p_d) \in \Theta$ le vecteur des paramètres.

En négligeant les dérivées temporelles, une équation aux dérivées partielles (EDP) paramétrique générique peut être exprimée comme suit :

$$\mathcal{L}(u(\boldsymbol{x},t);\boldsymbol{p}) = f(\boldsymbol{x},t) \tag{1}$$

où $\mathcal{L}(\bullet)$ fait référence à un opérateur différentiel générique (éventuellement non linéaire) impliquant des dérivées dans l'espace et présentant une dépendance paramétrique par rapport au vecteur p. De plus, f est le terme source et u est la solution du problème (ces fonctions peuvent éventuellement être vectorielles).

Le problème (1) est équipé d'une condition initiale appropriée spécifiant l'état initial de u et de conditions aux limites caractérisant son comportement à la frontière du domaine $\partial\Omega$, définissant ainsi un problème de valeurs aux limites (PVL). La nature spécifique de (1) dépend du problème physique ou du système particulier qui est modélisé. Des ouvrages tels que [17–19] fournissent une large gamme d'exemples et une base théorique solide sur de tels problèmes.

De nombreuses techniques ont été développées dans le cadre de l'analyse numérique des EDP [20], permettant des solutions de haute fidélité pour une grande variété de problèmes différentiels, avec des bornes d'erreur certifiées. Des exemples de ces méthodes sont les différences finies [21], les éléments finis [22], les volumes finis [23], les éléments de frontière [24], les méthodes spectrales [25] ou l'analyse isogéométrique [26].

Considérons d'abord les paramètres p comme fixes. En général, lors de la résolution de (1) numériquement, la solution est approximée comme suit :

$$u(\boldsymbol{x},t) \approx \sum_{i=1}^{N_{\boldsymbol{x}}} \psi_i(\boldsymbol{x}) u_i(t)$$
(2)

où N_x est le nombre de points de maillage dans tout le domaine spatial Ω , $\psi_i(x)$ sont des fonctions de forme spatiales choisies de manière opportune et $u_i(t)$ sont les valeurs

nodales inconnues. L'approximation (2) est injectée dans le problème (1) (plus précisément dans sa formulation faible), ce qui conduit à un système algébrique dépendant du temps d'équations (linéaires ou non linéaires). Ensuite, une méthode de marche dans le temps est introduite et les systèmes sont résolus de manière incrémentielle (si des dérivées temporelles apparaissent également, une attention particulière doit être portée au choix de la méthode temporelle).

Des expansions séparées similaires sont promues dans les modèles réduits classiques. Par exemple, en considérant la décomposition orthogonale aux valeurs propres (POD) [27,28], l'approximation est donnée par

$$u(\boldsymbol{x},t) \approx u^{\text{POD}}(\boldsymbol{x},t) = \sum_{k=1}^{M} w_k^{\boldsymbol{x}}(\boldsymbol{x}) \alpha_k^t(t), \qquad (3)$$

où les fonctions $w_k^x(x)$ et $\alpha_k^t(t)$, pour k = 1, ..., M, sont les modes spatiaux et temporels, qui forment des bases réduites de fonctions spatiales et temporelles. De plus, les modes POD satisfont une propriété d'orthogonalité qui garantit l'optimalité de la décomposition.

Les bases réduites POD regroupent la structure la plus typique ou caractéristique du système, réduisant la dimension de plusieurs ordres de grandeur par rapport aux modèles numériques traditionnellement employés. Cependant, l'inconvénient est que cette approximation est calculée a posteriori, c'est-à-dire une fois que le problème (1) est résolu au moyen de techniques habituelles.

La décomposition propre généralisée (PGD) [29,30] est une généralisation de la POD qui permet le calcul de représentations séparées similaires a priori, c'est-à-dire pendant la résolution du problème (1). Dans ce contexte, l'approximation supposée est écrite comme suit

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t), \qquad (4)$$

où les fonctions $U_k^{\boldsymbol{x}}(\boldsymbol{x})$ et $U_k^t(t)$, pour $k = 1, \ldots, m$, sont les modes spatiaux et temporels inconnus, qui ne sont généralement pas orthogonaux par rapport à ceux du POD. La solution (4) est construite de manière itérative par un enrichissement modal successif. En cherchant l'enrichissement m, les termes précédents m-1 sont connus, donc $u_m(\boldsymbol{x}, t)$ s'exprime comme

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^{m-1} U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) + U_m^{\boldsymbol{x}}(\boldsymbol{x}) U_m^t(t),$$
(5)

et est injectée dans le problème (1). La séparation des variables est exploitée au niveau des opérateurs différentiels pour construire des problèmes différentiels spatiaux et temporels séparés, pour $U_m^{\boldsymbol{x}}(\boldsymbol{x})$ et $U_m^t(t)$, respectivement. Une séquence de problèmes spatiaux et temporels séparés est résolue jusqu'à ce qu'un point fixe soit atteint pour le produit $U_m^{\boldsymbol{x}}(\boldsymbol{x})U_m^t(t)$. Cette technique est connue sous le nom de stratégie de direction alternée, où les problèmes séparés sont abordés via des différences finies ou des éléments finis standard [31].

Pour avoir une idée des économies computationnelles, considérons un modèle transitoire standard défini dans un espace physique en 3D. Impliquant k pas de temps, les stratégies incrémentales habituelles doivent résoudre k problèmes 3D (généralement non linéaires). En laissant p être le nombre d'itérations non linéaires nécessaires pour calculer chaque terme de la somme finie, il faudrait résoudre environ mp problèmes 3D pour calculer les fonctions spatiales et mp problèmes 1D pour calculer les fonctions temporelles. En général, mp est plusieurs ordres de grandeur inférieur à k. De plus, selon le problème analysé, d'autres décompositions peuvent être imposées pour obtenir des gains computationnels encore plus remarquables (par exemple, l'espace peut être entièrement ou partiellement séparé).

Un grand avantage des représentations basées sur la PGD est la possibilité d'inclure les paramètres en tant que coordonnées supplémentaires du problème dans la décomposition, permettant de construire efficacement des modèles de grande dimension, qui peuvent être considérés comme des vade-mecums computationnels ou des abaques numériques [31–35]. Dans ce cas, la solution s'exprime comme

$$u_m(\boldsymbol{x}, t, \boldsymbol{p}) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) U_k^{p_1}(p_1) \cdots U_k^{p_d}(p_d),$$
(6)

où $U_k^{p_j}$ sont des fonctions univariées du paramètre p_j .

Les modèles paramétriques basés sur la PGD reposent sur une phase hors ligne potentiellement coûteuse où un problème de grande dimension est résolu, et une phase en ligne où des évaluations efficaces du modèle sont effectuées, permettant une simulation en temps réel.

Ci-dessous, nous présentons brièvement les représentations séparées adoptées dans ce travail, qui sont séparées spatialement, temporellement et paramétriquement pour l'espace, le temps et les paramètres, respectivement.

ESPACE

Considérons un domaine en forme de plaque $\Omega = \Omega_{xy} \times \Omega_z$ avec $\Omega_z = [-h/2, h/2]$, où l'épaisseur h est inférieure aux dimensions dans le plan. Pour contourner les problèmes précédemment discutés liés à une discrétisation entièrement tridimensionnelle de telles géométries dégénérées, des méthodes efficaces développées dans le cadre de la PGD sont basées sur l'expression d'une fonction d'espace sous une forme séparée dans le plan/hors du plan

$$u_m(\boldsymbol{x}) = \sum_{k=1}^m U_k^{xy}(x, y) U_k^z(z).$$
 (7)

Dans ce contexte, les solutions 3D (N_x degrés de liberté) sont calculées en alternant des séries de problèmes 2D (N_{xy} degrés de liberté) et 1D (N_z degrés de liberté), préservant le coût computationnel d'une simulation 2D, tout en permettant une représentation détaillée et précise de la solution le long de la coordonnée d'épaisseur. La figure 3 montre la décomposition du domaine de calcul 3D en grilles 2D dans le plan et 1D hors du plan, entraînant une réduction importante du nombre initial de degrés de liberté.

Les représentations séparées dans l'espace telles que (7) ont été appliquées avec succès en mécanique des structures [36–42], pour la simulation d'écoulements de fluides newtoniens et non newtoniens se produisant dans des laminés composites et multiaxiaux, dans des inserts composites thermoplastiques et dans des polymères minces rugueux [43–46], pour des écoulements dans des milieux poreux stratifiés [47], pour des écoulements viscoplastiques non linéaires [48] et pour des processus micro-ondes dans



FIGURE 3: Décomposition d'une grille de calcul 3D dans une géométrie en forme de plaque.

des laminés composites à plis minces [38, 49, 50]. Pour avoir une idée des gains computationnels, on peut consulter, par exemple, [38] où plusieurs exemples numériques sont discutés.

Cependant, malgré l'efficacité de telles méthodes, leur applicabilité à des problèmes à grande échelle est souvent limitée en raison de leur nature intrusive. De nombreuses tentatives ont été menées dans cette direction pour proposer des solutions PGD moins intrusives, facilitant leur implémentation dans un logiciel commercial.

Dans [40], une approche plan/hors-plan peu intrusive est proposée pour l'élastoplasticité, permettant l'intégration de descriptions entièrement 3D dans les logiciels de plaques existants. De plus, une approche algébrique non intrusive plan/horsplan a été développée dans [51] et appliquée à la modélisation mécanique de tissus 3D. La méthodologie a ensuite été appliquée avec succès en élastodynamique paramétrique [52] et pour la simulation mécanique haute résolution d'une cellule de batterie lithium-ion stratifiée [53].

Dans ce travail, nous proposons une nouvelle approche plan/hors-plan moins intrusive pour la mécanique des structures, basée sur la combinaison des théories de coques (d'ordre un et d'ordre supérieur) avec un cadre d'apprentissage automatique.

TEMPS

Considérons l'intervalle de temps $I = (0, T_f)$ avec $T_f \gg 0$. Pour la simulation numérique de phénomènes à long terme nécessitant un pas de temps très petit ou pour des phénomènes présentant plusieurs échelles de temps, une représentation séparée dans le temps peut être adoptée. Cela peut être fait via une représentation séparée espace-microtemps-macrotemps [54,55]

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) \approx \sum_{k=1}^M U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^{\tau}(\tau) U_k^T(T),$$
(8)

ou en imposant la décomposition multi-temps pour le calcul de la fonction temporelle (sous-modes micro/macrotemps) [55, 56]

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) \approx \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j=1}^{m_k} U_{k,j}^{\tau}(\tau) U_{k,j}^T(T),$$
(9)

où deux nouvelles coordonnées temporelles indépendantes τ (microtemps, ou temps rapide) et T (macrotemps, ou temps lent) doivent être introduites. De plus, dans l'équation (8), le nouveau nombre de modes est noté M (en général, M > m), tandis que dans l'équation (9), le nombre de modes impliqués dans la séparation espacetemps est toujours m et m_k désigne le nombre de sous-modes temporels nécessaires pour approximer la fonction à échelle unique pour le mode global actuel k. La figure 4 montre le maillage temporel fin défini le long de la variable temporelle habituelle t, obtenu à partir d'un maillage fin le long de la variable micro-échelle rapide τ et d'un maillage grossier le long de la variable macro-échelle lente T. Si une approche standard de marche dans le temps nécessite une discrétisation de l'intervalle de temps I au moyen de N_t pas de temps, dans l'approche multi-temps, cela se recouvre comme $N_t = N_{\tau}N_T$ en mappant N_{τ} micro-degrés de liberté sur une grille plus grossière de N_T intervalles macro.



FIGURE 4: Décomposition d'un maillage temporel fin à une échelle en un maillage à plusieurs échelles.

Dans ce contexte, une solution à petite unique $(N_t \text{ degrés de liberté})$ est calculée en alternant des séries de problèmes à petite échelle (dimension N_{τ}) et de problèmes à grande échelle (dimension N_T), faisant passer le coût computationnel et les besoins en mémoire de $N_t = N_{\tau}N_T$ à $N_{\tau} + N_T$.

Jusqu'à présent, cette décomposition n'a été appliquée qu'à des cas linéaires, tels que des problèmes thermiques et élastodynamiques à plusieurs échelles [54,55]. Dans cette thèse, la procédure est étendue au cadre non linéaire de l'élasto-plasticité. De plus, pour permettre la simulation de problèmes à très long terme, la formulation est couplée à un cadre d'apprentissage automatique le long de la macro-échelle.

PARAMÈTRES

Considérons le domaine paramétrique $\Theta \subset \mathbb{R}^d$, avec $d \gg 1$. La PGD permet de calculer une solution paramétrique multivariée en traitant les paramètres comme des coordonnées supplémentaires du problème et en considérant une représentation séparée des paramètres

$$u_m(\boldsymbol{x}, t, \boldsymbol{p}) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) U_k^{p_1}(p_1) \cdots U_k^{p_d}(p_d).$$
(10)

Même si la dimension du problème augmente considérablement, la forme séparée de la solution PGD conduit à de petits problèmes indépendants dans chaque direction paramétrique, dont les solutions sont des fonctions paramétriques univariées (ou modes). Cet aspect puissant, associé à ses nombreuses applications (telles que l'identification inversée a posteriori et l'optimisation), est discuté dans la revue [57]. De plus, des études récentes combinent le solveur paramétrique basé sur la PGD avec des descriptions géométriques basées sur les NURBS, permettant d'intégrer efficacement également des paramètres géométriques complexes dans la décomposition [58, 59, 59, 60].

Grâce à la capacité à traiter des espaces de grande dimension, de telles approximations séparées suscitent un grand intérêt dans l'industrie, et plusieurs variantes non intrusives de la PGD ont été développées pour des études paramétriques [61–69]. Dans ce sens, la PGD sert de méthode de réduction de modèle basée sur des snapshots de simulation et devient un outil puissant pour effectuer des régressions non linéaires multiparamétriques avec peu de données.

Indépendamment de la nature intrusive ou non intrusive de la PGD, la création d'un modèle unique dans de grands domaines physiques et paramétriques est une tâche délicate. Une voie précieuse pour améliorer la précision consiste à partitionner l'espace physique afin de construire un modèle dans chacun des patchs résultants. En général, les modèles quasi linéaires locaux performent mieux que les modèles non linéaires riches dans l'ensemble du domaine spatial.

Ainsi, les schémas partitionnés sont reformulés dans un cadre paramétrique basé sur la PGD [70–72]. Dans ce contexte, le modèle paramétrique est construit via une procédure de décomposition de domaine avec une double perspective. D'une part, traiter des systèmes locaux plus petits réduit la complexité d'un problème global. D'autre part, cela améliore les perspectives physiques des modèles qui deviennent de plus en plus complexes et corrélés en dimensions élevées.

Dans cette thèse, nous proposons et étudions une procédure générale pour construire des modèles paramétriques dans le contexte de systèmes multi-composants. Cela repose sur la séparation du domaine en composants non chevauchants, la modélisation du squelette des interfaces et la construction de sous-problèmes locaux par composant et de substituts paramétriques peu coûteux. Ceci est illustré de manière schématique dans la figure 5.



FIGURE 5: Décomposition d'un problème paramétrique unique \mathcal{P} en sous-problèmes par composant \mathcal{P}_p .

CONTENU DU MANUSCRIT

Ce manuscrit est structuré en trois chapitres, alignés sur les trois défis abordés de l'espace, du temps et des paramètres.

CHAPITRE 2 : ESPACE

Dans la section 2.2, la théorie des coques du premier ordre est décrite, mettant l'accent sur ses limitations. Dans la section 2.3, la théorie améliorée proposée est présentée, en accordant une attention particulière aux détails de mise en œuvre dans un solveur commercial. Dans la section 2.4, quelques exemples numériques sont discutés. Enfin, la section 2.6 donne une brève conclusion et des perspectives.

CHAPITRE 3 : TEMPS

Le chapitre est divisé en deux sections. La section 3.2 est consacrée au calcul d'une réponse multi-échelle en temps en élasto-plasticité cyclique. La section 3.3 exploite les résultats de la section 3.2 pour construire un modèle efficace des relations constitutives non linéaires. Les sections sont structurées comme suit.

- 1. Section 3.2 : La sous-section 3.2.1 présente l'énoncé du problème dans ses formes forte et faible. La sous-section 3.2.2 reformule le problème dans le cadre numérique de la Décomposition Propre Généralisée, en commençant par la séparation de l'espace et du temps, puis en abordant la séparation multi-temps. La sous-section 3.2.3 présente les résultats des essais de traction charge-décharge. La sous-section 3.2.4 donne des conclusions et des perspectives.
- 2. Section 3.3 : La sous-section 3.3.1 rappelle brièvement le cadre théorique et numérique déjà introduit dans la section 3.2. La sous-section 3.3.2 entre dans les détails de toutes les méthodes utilisées pour construire le modèle basé sur les données. La sous-section 3.3.3 présente les résultats numériques en considérant deux tests de référence en 2D, en faisant varier la géométrie et la loi de plasticité. Enfin, la sous-section 3.3.4 fournit des conclusions et des perspectives.

Chapitre 4 : Paramètres

Le chapitre est structuré comme suit. La section 4.2 décrit toutes les étapes de la procédure proposée : (1) définition du squelette des interfaces ; (2) identification des paramètres du modèle local ; (3) identification des conditions de transmission ; (4) construction du modèle réduit local ; (5) imposition de l'équilibre du squelette des interfaces. La section 4.3 présente des applications à deux problèmes de référence. La première exemple concerne un problème de conduction en régime permanent. La deuxième exemple concerne l'état mécanique d'une plaque élastique mince. La section 4.3 donne des remarques conclusives et des perspectives.

Contents

1 Introduction			1	
	1.1	Context and motivations		1
		1.1.1	Space	2
		1.1.2	Time	4
		1.1.3	Parameters	4
	1.2	Separa	ated representations	5
		1.2.1	Space	7
		1.2.2	Time	8
		1.2.3	Parameters	9
	1.3	Manus	script content	10
		1.3.1	Chapter 2: Space	10
		1.3.2	Chapter 3: Time	10
		1.3.3	Chapter 4: Parameters	11
	1.4	Contr	ibutions to automotive engineering	11
		1.4.1	Vehicle crashworthiness, safety and optimization	11
		1.4.2	Safety aspects of lithium-ion batteries in electric vehicles	14
	1.5	Scient	ific contributions	16
		1.5.1	Publications in international journals	16
		1.5.2	Conferences proceedings	17
2	Spa	ce		19
	2.1	Introd		20
	2.2	Shell t	theory	22
	2.3	Enhar	nced theory	25
		2.3.1	Finite element formulation of the in-plane problem	26
		2.3.2	Out-of-plane kinematics estimation	30
2.4		3D sir	nulations	31
		2.4.1	Test case 1	31
		2.4.2	Test case 2	32
		2.4.3	Test case 3	34
		2.4.4	Test case 4	36

		2.4.5	Test case 5	38
	2.5	Machi	ine learning based training \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots	39
	2.6	Concl	usions	40
3	Tim	ıe		41
	3.1	Introd	luction	42
	3.2	Multi-	-time PGD in cyclic elasto-plasticity	43
		3.2.1	Theoretical framework	43
		3.2.2	Material and methods	47
		3.2.3	Results and discussion	52
		3.2.4	Conclusions	58
	3.3	Multi-	-time based learning of history-dependent nonlinear behaviors	59
		3.3.1	Theoretical and numerical framework	59
		3.3.2	Multiscale-based data-driven modeling	64
		3.3.3	Results and discussion	71
		3.3.4	Conclusions	79
	3.4	Concl	usions	80
4	Par	rameters		
	4.1	Introd	luction	82
	4.2	Mater	ials and methods	84
		4.2.1	NURBS-based geometry mapping and PGD-based parametric solutions	85
		4.2.2	Multi-parametric modularization	87
		4.2.3	Computational work-flow for online real-time simulations	92
	4.3	Result	ts and discussion	93
		4.3.1	Steady state heat conduction	93
		4.3.2	First order plate deformation	96
	4.4	Concl	usions	98
	Con	clusio	ns	99
\mathbf{A}	Imp	lemen	atation aspects of PGD, G-POD and HODMD	101
	A.1	Tenso	r-based PGD methods	101
	A.2	POD	and Gappy-POD	104
		A.2.1	POD	104
		A.2.2	Gappy-POD	104
	A.3	HODI	MD	105

В	Implementation aspects of the NURBS-PGD method		109
	B.1	NURBS-based geometry description	109
	B.2	Separation of space: separated approximate representation $-{\rm SAR-}$	110
	Bib	liography	113
	List of Figures		131
	List	of Tables	135

INTRODUCTION

Contents

1.1 Con	text and motivations $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 1$
1.1.1	Space
1.1.2	Time
1.1.3	Parameters 4
1.2 Sepa	arated representations
1.2.1	Space
1.2.2	Time
1.2.3	Parameters
1.3 Mar	$nuscript \ content \ \ldots \ 10$
1.3.1	Chapter 2: Space $\dots \dots \dots$
1.3.2	Chapter 3: Time $\dots \dots \dots$
1.3.3	Chapter 4: Parameters
1.4 Con	tributions to automotive engineering $\ldots \ldots \ldots 11$
1.4.1	Vehicle crashworthiness, safety and optimization $\ldots \ldots \ldots 11$
1.4.2	Safety aspects of lithium-ion batteries in electric vehicles 14
1.5 Scie	ntific contributions 16
1.5.1	Publications in international journals 16
1.5.2	Conferences proceedings $\dots \dots \dots$

1.1 CONTEXT AND MOTIVATIONS

Space, time, and parameters are the three basic variables describing any physical system. The space refers to the region in which the physical objects and phenomena are observed. The time governs the evolution of the different processes. The parameters are essential for building mathematical models of the physical phenomena, accounting for the specific properties and characteristics of the analyzed systems. Therefore, these

variables play a key role in simulation-based engineering [1-3], where reliable models require accurate descriptions of the spatial region, time evolution, and adequate parameter sensitivity analyses.

In numerical simulations [4,5], the spatial domain is typically discretized into elements (i.e., solid elements when considering a three-dimensional space) or grid points, defining the so-called computational mesh. Similarly, the time domain is discretized into small time steps. The governing equations and models are then transferred into their discrete counterparts, suitable for numerical computing via the so-called finite element method (FEM) and its variants. The parametric domain is explored solving the equations for different values of the parameters and observing the resulting changes in the system's response. It is well known that the mesh size affects the accuracy and stability of the simulation, but also the computing time since finer meshes yield higher computational costs. This also happens in parametric analyses, where a higher number of parameters, or larger parameter ranges, require longer computing time to explore the parametric domain.

In fact, despite the progress in numerical analysis and high-performance computing, many real-world problems and large-scale systems can lead to cost-prohibitive simulations. Model order reduction (MOR) [6,7] is the branch of computational science which faces this matter, seeking the right balance between simulation quality and efficiency. This thesis aims at enriching the current state-of-the-art MOR techniques on three levels, that is 1. space, 2. time and 3. parameters, motivated by three different challenging scenarios encountered nowadays in simulation-based engineering.

In space, the work focuses on the necessity of an efficient simulation framework valid from thin to thick geometries in structural mechanics. In time, the challenge is combining a very long duration of the phenomena with very small time step, for complex nonlinear problems in solid mechanics. Finally, as concerns the parameters, the work focuses on multi-component design problems involving a high-number of modeling parameters.

Each challenge is therefore motivated by a characteristic dimension in the system. For the space, such dimension is the thickness of the structure. For the time, it is the number of loading cycles or the duration of the simulation. For the parameters, this stands in the number of parameters.

The motivations of such three axes are exposed in more detail here below, omitting an extensive overview and description of most recent research and literature on the topics, which will be provided afterwards in each related chapter.

1.1.1 SPACE

The solution of 3D models in degenerated geometries, where one characteristic dimension is lower then others, is not trivial. This is the case of plates and shells, where the thickness (often referred as out-of-plane dimension) is smaller compared to its other dimensions (referred as in-plane ones) [8]. To cite a few examples of such structures, in engineering, they are encountered in automobile, aircraft or ship components and bodies, in micro-elecromechanical devices or electric batteries, in pipelines and tanks. In architecture, they are often parts of bridges, buildings and domes. In nature, one can simply imagine eggs, mollusks, skeletons, biologic membranes or the human skin.

When discretizing such a degenerated domain, the aspect ratio of the mesh elements (i.e., the ratio of the element's longest side to its shortest side) can become highly This property limits the applicability of traditional FEM based on 3D solid elements, leading to the necessity of developing specialized elements, known as 2D shell elements [9, 10]. These are based on the so-called shell theory, which makes some assumptions about the through-the-thickness behavior. In the same way, long and slender structures are usually discretized using the so-called 1D beam elements, relying on the beam theory.

Despite their effectiveness in many scenarios, shell elements also have several drawbacks: (a) they are primarily designed to analyze in-plane behavior, and they may not perform well in situations where out-of-plane deformations or buckling effects are significant; (b) shell elements are typically used for linear analyses, and they may not be suitable for nonlinear analyses, such as nonlinear elasticity due to material or geometrical nonlinearities, elastoplastic behaviors or complex multi-physics behaviors; (c) the basic shell theory is developed for thin-shell structures and results may be compromised when switching from thin to moderately-thick or thick-shell structures, where higher-order formulations and specific thick shell elements are required (usually the choice of the right element type is not evident).

Nevertheless, classic shell elements still represent the most common choice in industry, due to their efficiency. To give an idea, in full vehicle, we can count approximately 10^2 beam elements, 10^6 shell elements and 10^4 solid elements. One method currently used to get around the problems mentioned above is to execute, during the simulation, a shell-to-solid remeshing (SSR) for some specific parts, when the results start appearing doubtful and when more intricate physics effects need to be captured. This requires an extrude of the part followed by a remeshing involving a high number of solid elements, which significantly increases the computing time. Moreover, this procedure also entails other difficulties such as the coupling of the extruded part with the surrounding 1D beam elements or 2D shell elements, which is usually not trivial.

If we consider the body in white (BIW) structure, almost all pieces are thin structures, as shown in the figure 1.1, where three of them are highlighted in orange.



FIGURE 1.1: Examples of thin structures in the body in white (courtesy of ESI group).

In this context, the main objective of the thesis is to suggest and investigate alternative routes for an efficient simulation framework valid from thin to thick shell structures.

1.1.2 TIME

Despite the availability of high performance computing platforms, the numerical solution of complex, time-dependent, non-linear problems still nowadays may be a cumbersome challenge. This issue is even acerbated in case of multiscale behaviors, where effects occur at different space and time scales. In such contexts, the treatment of the different scales of the problem can be compulsory to define reliable models. Such challenges are encountered in computational solid mechanics (CSM) [11], when dealing with metamaterials, additive manufacturing, sheet metal forming, cyclic viscoplasticity or dynamics with loadings involving multiple characteristic times.

As a major challenge, this situation becomes extremely delicate when the phenomena shall be simulated over very large time intervals and, at the same time, the system response must encompass the different time scales present in the model. Standard discretization techniques are constraint to mesh up to the finest scale to predict accurately the response of the system, resulting in a dramatic increase in computational cost and, in some cases, becoming computationally infeasible. Additionally, such scenarios entail other issues such as the accumulation of numerical round-off errors, the saturation of storage resources or the requirement of adaptive time-steps.

Let us take the case of fine-scale cyclic fatigue simulations, where fatigue damage accumulates gradually with each cycle. To give some order of magnitude, the number of cycles to failure in fatigue life amounts to 10^4 - 10^5 in the case of low cycle fatigue (LCF), 10^5 - 10^7 for high cycle fatigue (HCF), and more then 10^7 for very high cycle fatigue (VHCF). For instance, most of the load-bearing applications in vehicles, engine parts, are loaded with 10^8 cycles, while railway components, bridges and wheels are loaded with 10^9 cycles in their lifetime [12, 13]. Moreover, for a fine-scale simulation, 10^3 time steps could be necessary over a single cycle. These figures directly show that simulating every individual cycle is infeasible.

Instead of simulating all single cycles, other techniques are employed, such as statistical approaches, damage accumulation models, simplification techniques such as rainflow-counting algorithm or the Miner's rule [14-16].

In this sense, the thesis aims at proposing a new time multiscale procedure and investigating its effectiveness in nonlinear problems.

1.1.3 PARAMETERS

Most of engineering applications and processes require parametric studies, where the parametric dependency can be of both physical (e.g., model coefficients, source terms) and geometric (e.g., domain shape) nature. This happens, for instance, in optimization, uncertainty quantification and propagation, inverse identification or simulation-based control. In all these many-query scenarios, almost real-time evaluations of the system response are needed. Despite the advances in the model order reduction (MOR) and machine learning (ML) [6,7] to build parametric surrogates, the task is still challenging when dealing with in large domains, large-scale systems, combined with a high number of parameters.

In fact, as the number of parameters increases, the problem's dimensionality grows exponentially (curse of dimensionality) and a large number of simulations, each involving different parameter combinations, are required to explore the parameter space, straining computational resources, memory, and storage. Moreover, in a high-dimensional parameter space, parameters may be interdependent, making it challenging to identify the most relevant combinations to study. This can lead to the exploration of redundant or uninformative parameter sets. Other limitations are the identification of optimal parameter sets, the study of parameter sensitivities, but also the model interpretability and validation itself.

This issue is typically encountered in multi-component parametric structures, where various components or parts interact and their behaviors depend on different parameters. To give an idea of the difficulties, we take again an industrial problem in automotive engineering. The construction of a parametric crash solution of a structure composed of 200 parts, each one involving a parametric material described by 3 parameters and the part thickness. Thus, the number of parameters becomes exorbitant, 800. In the most advanced regression technologies the number of data (high-resolution problem solutions) scales with the number of parameters. In the present case, even by using the most advanced regression techniques more than 800 high-fidelity solutions seems compulsory, but considering that each of these solutions implies one-day computation, the computational cost represents 3 years calculations.

Figure 1.2 exemplifies the challenge, highlighting some possible parametric components in the vehicle B-pillar.



FIGURE 1.2: Example of a multi-parametric structure in automotive (courtesy of Hyundai Motor Group).

1.2 Separated representations

Let us consider a generic differential problem defined over the spatial region $\Omega \subset \mathbb{R}^n$ (with n = 1, 2, 3), evolving within the time interval $I \subset \mathbb{R}$ and incorporating parameters residing in the parametric space $\Theta \subset \mathbb{R}^d$ (with $d \in \mathbb{N}$). Let $\mathbf{x} = (x, y, z) \in \Omega$ be the vector of spatial coordinates, $t \in I$ the time coordinate and $\mathbf{p} = (p_1, \ldots, p_d) \in \Theta$ the vector of parameters.

Neglecting the time derivatives, a generic parametric partial differential equation (PDE) can expressed as

$$\mathcal{L}(u(\boldsymbol{x},t);\boldsymbol{p}) = f(\boldsymbol{x},t) \tag{1.2.1}$$

where $\mathcal{L}(\bullet)$ refers to a generic (eventually nonlinear) differential operator involving the derivatives in space and exhibiting a parametric dependency upon the vector p. Moreover, f is the source term and u is the problem solution (these can eventually be vector-valued functions).

Problem (1.2.1) is equipped with a suitable initial condition specifying the initial state of u and boundary conditions characterizing its behavior at the boundary of the domain $\partial\Omega$, defining a so-called boundary value problem (BVP). The specific nature of
1.2.1 depends on the particular physical problem or system being modeled. Textbooks such as [17–19] provide a wide range of examples and a solid theoretical background about such problems.

Many techniques have been developed in the framework of numerical analysis of PDEs [20], allowing high-fidelity solutions of a wide variety of differential problems, with certified error bounds. Examples of these methods are finite differences [21], finite elements [22], finite volumes [23], boundary elements [24], spectral methods [25] or isogeometric analysis [26].

Let us first keep the parameters p fixed. Generally, when solving (1.2.1) numerically, the solution is approximated as

$$u(\boldsymbol{x},t) \approx \sum_{i=1}^{N_{\boldsymbol{x}}} \psi_i(\boldsymbol{x}) u_i(t)$$
(1.2.2)

where N_x is the number of mesh points throughout the spatial domain Ω , $\psi_i(x)$ are opportunely chosen spatial shape functions and $u_i(t)$ the unknown nodal values. Approximation (1.2.2) is injected into problem (1.2.1) (more precisely in its weak formulation) leading to a time-dependent algebraic system of (linear or nonlinear) equations. Afterwards, a time marching scheme is introduced and the systems are solved incrementally (if time derivatives also appear, special care must be paid to the choice of the time scheme).

Similar separated expansions are promoted in classical reduced-order models. For instance, when considering the proper orthogonal decomposition (POD) [27, 28], the approximation reads

$$u(\boldsymbol{x},t) \approx u^{\text{POD}}(\boldsymbol{x},t) = \sum_{k=1}^{M} w_k^{\boldsymbol{x}}(\boldsymbol{x}) \alpha_k^t(t), \qquad (1.2.3)$$

where the functions $w_k^x(\boldsymbol{x})$ and $\alpha_k^t(t)$, for $k = 1, \ldots, M$, are the space and time modes, which form low-dimensional reduced bases of spatial and time functions. Moreover, the POD modes satisfy an orthogonality property which guarantees the optimality of the decomposition.

The POD reduced bases gather the most typical or characteristic structure of the system, reducing the dimension of several order of magnitudes compared to traditionally employed numerical models. However, the inconvenient is that such approximation is computed a posteriori, that is only once problem (1.2.1) is solved by means of usual techniques.

The proper generalized decomposition (PGD) [29, 30] is a generalization of the POD which enables the computation of similar separated representations a priori, that is while solving problem (1.2.1). In this context, the assumed approximation is written as

$$u_m(\boldsymbol{x}, t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t), \qquad (1.2.4)$$

where the functions $U_k^x(\mathbf{x})$ and $U_k^t(t)$, for $k = 1, \ldots, m$, are the unknown space and time modes, which are, in general, not orthogonal compared to the POD ones. Solution (1.2.4) is built iteratively by successive modal enrichment. When looking for the enrichment m, the previous m - 1 terms are known, thus $u_m(\mathbf{x}, t)$ is expressed as

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^{m-1} U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) + U_m^{\boldsymbol{x}}(\boldsymbol{x}) U_m^t(t), \qquad (1.2.5)$$

and injected into problem (1.2.1). The separation of variables is exploited at the level of the differential operators to build separated space and time differential problems, for $U_m^x(x)$ and $U_m^t(t)$, respectively. A sequence of separated space and time problems in solved until a fixed point is reached for the product $U_m^x(x)U_m^t(t)$. This technique is known as alternating direction strategy, where the separated problems are tackled via standard finite differences or finite elements [31].

To get an idea of the computational savings, let us consider a standard transient model defined in a 3D physical space. Involving k time steps, usual incremental strategies must solve k (in general nonlinear) 3D problems. Letting p be the number of nonlinear iterations needed for computing each term of the finite sum, one should solve around mp 3D problems for computing the space functions and mp 1D problems for computing the space functions and mp 1D problems for computing the time functions. In general, mp is many order of magnitudes lower than k. Moreover, depending on the analyzed problem, further decompositions can be enforced to obtain even more remarkable computational gains (for instance, the space can be fully or partially separated).

A great advantage of PGD-based representations is the possibility of including parameters as problem extra-coordinates within the decomposition, enabling to efficiently build high-dimensional models, which can be seen as computational vademecums or digital abacuses [31–35]. In this case the solution is expressed as

$$u_m(\boldsymbol{x}, t, \boldsymbol{p}) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) U_k^{p_1}(p_1) \cdots U_k^{p_d}(p_d), \qquad (1.2.6)$$

where $U_k^{p_j}$ are univariate functions of the parameter p_j .

PGD-based parametric models rely on a potentially expensive offline phase in which a high-dimensional problem is solved and an online phase where efficient evaluations of the model are performed, enabling real-time simulation.

Here below we quickly introduce the separated representations adopted in this work, which are space-separated, time-separated and parameters-separated for space, time and parameters, respectively.

1.2.1 Space

Let us consider a plate-like domain $\Omega = \Omega_{xy} \times \Omega_z$ with $\Omega_z = [-h/2, h/2]$, where the thickness h is lower than the in-plane dimensions. To circumvent the previously discussed issues related to a fully 3D discretization of such degenerate geometries, efficient methods developed in the PGD framework are based on expressing a function of space in an in-plane/out-of-plane separated form

$$u_m(\boldsymbol{x}) = \sum_{k=1}^m U_k^{xy}(x, y) U_k^z(z).$$
(1.2.7)

In this context, 3D solutions (N_x degrees of freedom) are computed alternating series of 2D (N_{xy} degrees of freedom) and 1D (N_z degrees of freedom) problems, preserving the computational cost of a 2D simulation, while enabling a detailed and accurate representation of the solution along the thickness coordinate. Figure 1.3 shows the decomposition of the 3D computational domain into the 2D in-plane and 1D outof-plane grids, entailing an important reduction of the original number of degrees of freedom.



FIGURE 1.3: Decomposition of a 3D computational grid in a plate-like geometry.

Space-separated representations such as (1.2.7) have been successfully applied in structural mechanics [36–42], for the simulation of squeeze flows of Newtonian and non Newtonian fluids occurring in composite and multiaxial laminates, in thermoplastic composite inserts and in thin rough polymers [43–46], to flows in stratified porous media [47], to nonlinear viscoplastic flows [48] and to microwave processes in thin-ply composites laminates [38, 49, 50]. To have an idea of the computational gains, one can check, for instance [38] where several numerical examples are discussed.

However, despite the effectiveness of such methods, their applicability to large-scale problems is often limited by their intrusive nature. Many attempts have been conducted in this direction to propose less intrusive PGD-based solutions, facilitating the implementation into a commercial software.

In [40] a minimally-intrusive in-plane/out-of-plane approach is proposed for elastoplasticity, allowing the integration of fully 3D descriptions into existing plate software. Moreover, a non-intrusive algebraic in-plane/out-of-plane approach was developed in [51] and applied to the mechanical modeling of 3D woven fabrics. The methodology has then successfully been applied in parametric elastodynamics [52] and for the highresolution mechanical simulation of a layered lithium-ion battery cell [53].

In this work we propose a novel less-intrusive in-plane/out-of-plane approach for structural mechanics, based on combining the shell theories (first-order and higher-order) with a machine learning framework.

1.2.2 TIME

Let us consider the time interval $I = (0, T_f)$ with $T_f \gg 0$. For the numerical simulation of long-term simulations requiring a really small time step or for phenomena exhibiting multiple time scales, a time-separated representation can be adopted. This can be done via a space-microtime-macrotime separated representation [54, 55]

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) \approx \sum_{k=1}^M U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^{\tau}(\tau) U_k^T(T), \qquad (1.2.8)$$

or imposing the multi-time decomposition for the computation of the time function (micro/macro time sub-modes) [55, 56]

$$u_m(\boldsymbol{x},t) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) \approx \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j=1}^{m_k} U_{k,j}^{\tau}(\tau) U_{k,j}^T(T), \quad (1.2.9)$$

where two new independent time coordinates τ (microtime, or fast time) and T (macrotime, or slow time) to be introduced. Moreover, in equation (1.2.8) the new number of modes is denoted as M (in general, M > m), while in equation (1.2.9) the number of modes involved in the space-time separation is still m and m_k denotes the number of time sub-modes required to approximate the single-scale function for the current global mode k.

Figure 1.4 shows the fine time grid defined along the usual time variable t obtained as from a fine grid along the fast micro-scale variable τ and a coarse grid along the slow macro-scale variable T. If a standard time-marching approach requires a discretization of the time interval I by means of N_t time steps, in the multi-time approach this is recovered as $N_t = N_\tau N_T$ by mapping N_τ micro-dofs onto a coarser grid of N_T macro intervals.



FIGURE 1.4: Decomposition of a single-scale fine time grid into a multi-scale one.

In this context, a fine-scale solution $(N_t \text{ degrees of freedom})$ is computed alternating series of micro-scale (dimension N_{τ}) and macro-scale problems (dimension N_T), making the computational cost and memory requirements switching from $N_t = N_{\tau}N_T$ to $N_{\tau} + N_T$.

So far this decomposition has been applied only in linear cases, such as thermal and elastodynamic multiscale problems [54,55]. In this thesis, the procedure is extended to the nonlinear setting of elasto-plasticity. Moreover, to enable the simulation of really long-term problems, the formulation is coupled with a machine learning framework along the macro-scale.

1.2.3 PARAMETERS

Let us consider the parametric domain $\Theta \subset \mathbb{R}^d$, with $d \gg 1$. The PGD allows to compute a multi-variate parametric solution treating the parameters as problem extra-coordinates and considering a parameters-separated representation

$$u_m(\boldsymbol{x}, t, \boldsymbol{p}) = \sum_{k=1}^m U_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) U_k^{p_1}(p_1) \cdots U_k^{p_d}(p_d).$$
(1.2.10)

Even if the dimensionality of the problem considerably increases, the separated form of the PGD solution conducts to small independent problems in each parametric direction, whose solutions are univariate parametric functions (or modes). This powerful aspect together with its many applications (such as a posteriori inverse identification and optimization) are discussed in the review [57]. Moreover, recent studies combine the PGD-based parametric solver with NURBS-based geometrical descriptions, allowing to efficiently integrate also complex geometric parameters in the decomposition [58, 59, 59, 60].

Thanks to the capability of dealing with high-dimensional spaces, such separated approximations have reached a great interest in industry and several non-intrusive PGD variants have been developed for parametric studies [61–69]. In this sense, the PGD serves as a snapshots-based model-reduction method and becomes a powerful tool to perform multi-parametric nonlinear regressions at the scarce-data limit.

Independently from the intrusive or non-intrusive nature of the PGD, creating a unique model in large physical and parametric domains is a tricky issue. A valuable route for enhancing accuracy consists in partitioning the physical space, in order to build a model in each of the resulting patches. Local quasi-linear models perform in general better than rich nonlinear ones in the whole space domain.

Partitioned schemes are thus being recast in a PGD-based parametric framework [70–72]. In this context, the parametric model is built via a domain-decomposition procedure with a double perspective. On the one hand, dealing with smaller local systems reduces the complexity of a single global problem. On the other hand, this improves the physics insights of the models which become the more and more intricate and correlated in high dimensions.

In this thesis, we propose and investigate a general procedure to build parametric models in the context of multi-component systems. This is based on separating the domain in non-overlapping components, modeling the interfaces skeleton and building local by-components and computationally cheap parametric surrogates. This is schematically illustrated in figure 1.5.



FIGURE 1.5: Decomposition of a single parametric problem \mathcal{P} into by-component subproblems \mathcal{P}_p .

1.3 MANUSCRIPT CONTENT

This manuscript is structured in three chapters, aligned with the three addressed challenges of space, time and parameters.

1.3.1 CHAPTER 2: SPACE

In section 2.2 the first-order shell theory is described, emphasizing its limitations. In section 2.3 the proposed enhanced theory is presented, paying attention to the implementation details in a commercial solver. In section 2.4 some numerical examples are discussed. Finally, section 2.6 gives a short conclusion and perspectives.

1.3.2 CHAPTER 3: TIME

The chapter is divided into two sections. Section 3.2 is dedicated to the computation of a time-multiscale response in cyclic elasto-plasticity. Section 3.3 exploits the results of section 3.2 to build an efficient data-driven model of the nonlinear constitutive relations. The sections are structured as follows.

- 1. Section 3.2. Subsection 3.2.1 presents the problem statement in its strong and weak forms. Subsection 3.2.2 recasts the problem in the numerical framework of the proper generalized decomposition, starting from the space and time separation and then addressing the multi-time separation. Subsection 3.2.3 shows the results on load-unload tensile tests. Subsection 3.2.4 gives conclusions and perspectives.
- 2. Section 3.3. Subsection 3.3.1 briefly recalls the theoretical and numerical framework already introduced in section 3.2. Subsection 3.3.2 enters in the details of all the methods exploited to build the data-driven model. Subsection 3.3.3 shows the numerical results considering two benchmark tests in 2D, varying the geometry and plasticity law. Finally, subsection 3.3.4 provides conclusions and perspectives.

1.3.3 Chapter 4: Parameters

The chapter is structured as follows. Section 4.2 describes all the steps of the proposed procedure: (1) definition of the interfaces skeleton; (2) identification of local model parameters; (3) identification od transmission conditions; (4) construction of local ROM; (5) imposition of the interface skeleton equilibrium. Section 4.3 shows applications two benchmark problems. The first example concerns a steady state conduction problem. The second example deals with the mechanical state of a thin elastic plate. Section 4.3 gives conclusive remarks and perspectives.

1.4 Contributions to automotive engineering

A significant part of this thesis has been devoted to the application of PGD-based model reduction tools in automotive engineering. These contributions, not covered by the manuscript, can be resumed in two areas: (1) vehicle crashworthiness, safety and optimization; (2) safety aspects of lithium-ion batteries in electric vehicles.

1.4.1 VEHICLE CRASHWORTHINESS, SAFETY AND OPTIMIZATION

In the automotive industry, building parametric surrogate models is a fundamental tool to evaluate, in real time, the performance of newly designed car components. Such models allow to compute any quantity of interest (QoI), such as a specific safety protocol index, for any choice of material and/or geometrical parameters characterizing the component, within the stringent real time constraint. For instance, they can be exploited to guarantee safer designs (e.g., maximizing energy absorption by the crash boxes) or to reduce manufacturing costs (e.g., minimizing the mass of a specific structure under some safety protocol constraints). In general, these parametric simulation tools allow a significant gain in terms of manufacturing costs and time delays during the investigation phase.

Non-intrusive PGD techniques [61] construct the reduced representation from high-fidelity solution snapshots obtained via any commercial software, requiring only post-processing steps such as interpolations into the parametric space. Thanks to their non-intrusive character and to the capability of handling a high-number of parameters while limiting the overfitting, sPGD-based surrogates have reached a great interest in industry [62–68].

In [63], we focus on the vehicle frontal structure system considering its performance in a full-frontal crash scenario. In the front structure system we parameterize the crash boxes (left and right) and the inner/outer side front members (left and right, front and rear) with respect to the part thickness and the material parameters, for a total number of 13 parameters. The structure is shown in figure 1.6.



FIGURE 1.6: Vehicle frontal structure under study (courtesy of Hyundai Motor Group).

The core of the work is considering material properties in the metamodel, such as the parameters characterizing the Krupkowski strain-hardening plasticity law. For this to be done, two important novelties are introduced. The first one concerns the design of experiments (DoE) and, particularly, a physics-informed sampling strategy following the manifolds experimentally observed. The second one is related to the strain-dependent plasticity accounted by using Neural Networks, respecting the static and dynamic tests experimentally performed over materials specimens. These points are exposed in more detail here below.

SAMPLING STRATEGY For each steel part, we consider the 3 parameters (n, K, ε_0) characterizing the Krupkowski strain-hardening law

$$\sigma = K(\varepsilon + \varepsilon_0)^n,$$

linking the true strength and the true strain.

To obtain physically-consistent results, the design of experiments -DoE- is based on three physical properties: the yield strength YS (R_p) , the ultimate tensile strength UTS (R_m) and the uniform elongation U-El $(A_g, \text{ in } \%)$. From the sampled tuple (YS, UTS, U-El), we compute the corresponding Krupkowski parameters (K, n, ε_0) by means of a non-linear optimization algorithm, as shown in figure 1.7, which shows the location of such points over a typical plasticity curve linking the engineering strength and engineering strain.

Available experimental data (in Hyundai Motor Group) collect the material properties observed (YS, UTS, U-El) for tests performed over specimens ranging from mild steel to press hardened steel (PHS). With the aim of sampling new materials close to the manifold of experimental data, we propose a sampling strategy based on the k-nearest neighbors (alternatives would be manifold learning techniques)



FIGURE 1.7: Plasticity curve (σ, ε) and location of sampled points.

STRAIN-RATE EFFECT Since strain-rate effect is also considered4, the material of a part is identified by a rate-dependent plasticity curve (i.e., a plasticity curve for each rate). Also in this case, an experimental dataset is available. This links material properties observed at quasi-static test (YS, UTS, UEI)^{QS} with the ones obtained at dynamic test under a strain rate r, (YS, UTS, UEI)^D.

After the sampling of a virtual material through its properties at quasi-static test, its response to strain-rate is predicted through trained neural-networks models over the available data, allowing to obtain a plasticity curve for each rate characterizing such newly defined material.

MULTI-SPGD Another element of novelty of this work is the usage of a multi-PGD approach to enhance the quality of the model. In fact, a clear bifurcation in the parametric space is observed, due to two different behaviors in the system response, that is buckling and compression modes.

The multi-PGD approach consists in clustering the high-fidelity solutions related to the considered sampling, for example by invoking the k-means [73] or a hierarchical cluster. Then, a nonlinear regression model \mathcal{M}_k is created from the solutions in each cluster k. Finally, the trickiest issue becomes the way of associating a cluster to any parameters choice, that is, performing an accurate classification.

The procedure, shown in figure 1.8, can be summarized in the following steps:

- 1. clustering high-fidelity solutions related to a design of experiments;
- 2. creating a regression model in each cluster (for instance, via the standard sPGD-based algorithms);
- 3. constructing a classifier able to associate a cluster to any parameters choice and to select the most suitable regression model.

When a parameter choice is in the border of two clusters, one could compute both regressions and then proceed to average them. Another procedure consists of rendering continuous the approximations of the different clusters by constructing a partition of unity on a mesh attached to the cluster centers of gravity. Another possibility consists in using mowing least squares in the regression construction like in the element-free Galerkin methods.



FIGURE 1.8: Multi-PGD procedure.

1.4.2 SAFETY ASPECTS OF LITHIUM-ION BATTERIES IN ELECTRIC VE-HICLES

In the conception and design phase of electric vehicles (EVs), additional safety requirements for batteries must be addressed. In face, a crucial safety concern is preventing internal short circuits (SCs) in battery cells resulting from damage during a crash, due to the associated fire hazard. However, evaluating SC risk in vehicle-level crash simulations is intricate due to phenomena occurring at various scales. The vehicle deforms on a macroscale level during impact, while battery cells locally deform, damaging the thin separator foil and potentially leading to an internal SC. This scales cascade is shown in figure 1.9.



FIGURE 1.9: Various length scales occurring in EV crash simulation (courtesy of ESI Group).

However, integrating detailed cell descriptions into crash simulations is impractical due to model size and the resulting very small stable timestep imposed by explicit time discretization. To get an idea of the complexity of such scales cascade, the number of elements needed for a detailed model of a single cell (meso-scale level) is around 5 millions, which is comparable to the number of elements required for the whole vehicle (macro-scale level). Considering that one module consists of around 16 cells and that one battery pack is composed of around 24 modules, this would bring to billions of elements only for the battery, which is unaffordable. In terms of time-steps the meso-scale cell model is around 10^{-3} µs, while the macro-scale vehicle model about 0.5 µs, implying major simulation challenges.

Model order reduction and machine learning have therefore reached an important place in the field of battery simulations and efficient PGD-based solutions have recently been proposed in [53, 64, 74].

DETAILED BATTERY CELL SIMULATION Lithium-ion cells can be considered as laminates of thin plies called anode, separator, and cathode [75]. The stack-up of such layers is often called jellyroll or jellystack and is shown in figure 1.10.



FIGURE 1.10: Detailed cell model structure (courtesy of ESI Group).

Detailed modeling approaches [76,77] consider the layers individually. Unfortunately, the usage of simplification hypotheses for thin structures, coming from shell theories, is limited since the cells are vulnerable towards out-of-plane loading (in fact, the internal short-circuit is mainly caused by the rupture of the separators). As a consequence, a rich description of out-of-plane strains and stresses is necessary and requires the employment of solid elements. This situation strongly impacts the computational efficiency, since the minimum element dimension is limited with the ply thickness (usual drawbacks arising when meshing thin structures with volume elements).

To reduce these computational issues, RVE models have been proposed, which limit the cell to a so-called representative volume element (RVE) or unit cell, and the entire mechanical behavior of the cell is determined with the help of homogenization theories [78,79]. However, the numerical simulation of a detailed unit cell via solidbased meshes can still be computationally expensive when using volume elements. In particular, this is experienced when employing implicit schemes, which requires the inversion of a big assembled liner system.

In [53], we suggest to overcome this issue in a non-intrusive manner, exploiting the PGD-based in-plane/out-of-plane decomposition to solve the algebraic system resulting from the finite element assembly, yielding important computational gains.

MACRO-MECHANICAL MODELS CALIBRATION Another class of homogenized models is constituted by the macro-mechanical ones [80–82]. The goal of this modeling is to obtain a computationally efficient model of the lithium-ion cell that reflects the mechanical behavior against different load cases. For this purpose, both 1D and 2D elements are used. The basic scheme of this modeling approach is shown in figure 1.11 and consists of three main components: pouch (blue), middle layer (green), and beams (red).

These models are extremely efficient. To have an idea of the computational gains, they allow to reduce from the 5 million elements and a 10^{-3} µs time-step of the meso-scale model (8 hours simulation on 64 CPUs) to 20 thousand elements and a time-step about



FIGURE 1.11: Macroscopic modeling approach (courtesy of Vehicle Safety Institute).

0.1 µs time-step (5 minutes simulation on 16 CPUs), making the battery simulation scalable with respect to the full vehicle crash.

However, these artificial material properties cannot be determined by component tests and must be calibrated by cell tests. This step can be time costly due to the number of calibration parameters and of test configurations to be examined, especially when proceeding experimentally. For instance, figure 1.12 shows three different configurations, where also the material and geometrical properties of the impactor can be varied.



FIGURE 1.12: Isometric view of four different test configurations (courtesy of Vehicle Safety Institute).

Parametric investigations have therefore reached a growing interest in this context and an efficient meta-modelling approach based on the sPGD and ANOVA-PGD is considered in [64, 74]. In these studies, a parametric model is built from high-fidelity finite elements simulations, varying the impactor properties (material and geometrical) for several loading cases. Afterwards, the metamodels allow the computation of system response in real-time and therefore enable the fast calibration of the macro-mechanical model against the experimental data.

1.5 Scientific contributions

1.5.1 Publications in international journals

- S. RODRIGUEZ, A. PASQUALE, K. NGUYEN, A. AMMAR, AND F. CHINESTA, A time multiscale based data-driven approach in cyclic elasto-plasticity, Computers & Structures, 295 (2024), p. 107277
- A. PASQUALE, S. RODRIGUEZ, K. NGUYEN, A. AMMAR, AND F. CHINESTA, *A time multiscale decomposition in cyclic elasto-plasticity*, Computers & Math-ematics with Applications, 149 (2023), pp. 75–83

- M.-J. KAZEMZADEH-PARSI, A. PASQUALE, D. DI LORENZO, V. CHAMPANEY, A. AMMAR, AND F. CHINESTA, Nurbs-based shape parametrization enabling pgd-based space separability: Methodology and application, Finite Elements in Analysis and Design, 227 (2023), p. 104022
- 4. A. SCHMID, A. PASQUALE, C. ELLERSDORFER, V. CHAMPANEY, M. RAFFLER, S. GUÉVELOU, S. KIZIO, M. ZIANE, F. FEIST, AND F. CHINESTA, Pgd based meta modelling of a lithium-ion battery for real time prediction, Frontiers in Materials (Computational Materials Science), 10 (2023)
- 5. A. SCHMID, A. PASQUALE, C. ELLERSDORFER, M. ZIANE, M. RAFFLER, V. CHAMPANEY, F. FEIST, AND F. CHINESTA, Application of pgd separation of space to create a reduced-order model of a lithium-ion cell structure, Frontiers in Materials, 10 (2023)
- 6. A. PASQUALE, V. CHAMPANEY, Y. KIM, N. HASCOËT, A. AMMAR, AND F. CHINESTA, A parametric metamodel of the vehicle frontal structure accounting for material properties and strain-rate effect: application to full frontal rigid barrier crash test, Heliyon, 8 (2022), p. e12397
- C. GHNATIOS, V. CHAMPANEY, A. PASQUALE, AND F. CHINESTA, A regularized real-time integrator for data-driven control of heating channels, Computation, 10 (2022)
- V. CHAMPANEY, A. PASQUALE, A. AMMAR, AND F. CHINESTA, Parametric curves metamodelling based on data clustering, data alignment, pod-based modes extraction and pgd-based nonlinear regressions, Frontiers in Materials, 9 (2022), p. 904707
- 9. A. PASQUALE, A. AMMAR, A. FALCO, S. PEROTTO, E. CUETO, J.-L. DUVAL, AND F. CHINESTA, A separated representation involving multiple time scales within the proper generalized decomposition framework, Advanced Modeling and Simulation in Engineering Sciences, 8 (2021), p. 26

1.5.2 Conferences proceedings

- A. SCHMID, A. PASQUALE, C. ELLERSDORFER, M. RAFFLER, V. CHAMPANEY, M. ZIANE, F. FEIST, AND F. CHINESTA, Mechanical characterization of li-ion cells and the calibration of numerical models using proper generalized decomposition, International Mechanical Engineering Congress and Exposition (IMECE), October 29-November 2 2023
- S. RODRIGUEZ, A. PASQUALE, N. KHANH, A. AMMAR, AND F. CHINESTA, *A time multiscale based data-driven approach in cyclic elasto-plasticity*, XVII International Conference on Computational Plasticity, Fundamentals and Ap-plications (COMPLAS), September 5-7 2023
- 3. A. PASQUALE, M. ZIANE, A. AMMAR, AND F. CHINESTA, Machine learning enhanced shell element formulation for the mechanical response of thin to thick structures, XVII International Conference on Computational Plasticity, Fundamentals and Applications (COMPLAS), September 5-7 2023

- 4. A. PASQUALE, D. DI LORENZO, V. CHAMPANEY, F. CHINESTA, E. CUETO, AND D. BAILLARGEAT, *Towards smart city digital twins*, International Congress on Industrial and Applied Mathematics (ICIAM), August 20-25 2023
- 5. D. DI LORENZO, V. CHAMPANEY, A. PASQUALE, AND F. CHINESTA, *Models correction based on sparse identification and data assimilation*, International Congress on Industrial and Applied Mathematics (ICIAM), August 20-25 2023
- 6. A. SCHMID, C. ELLERSDORFER, M. RAFFLER, F. FEIST, A. PASQUALE, V. CHAMPANEY, S. GUÉVELOU, M. ZIANE, AND F. CHINESTA, *Meta-modelling* of a li-ion battery for real time prediction based on proper generalized decomposition, Advanced Materials Poster Day (Tu Graz), February 2 2023
- 7. D. DI LORENZO, A. PASQUALE, V. CHAMPANEY, M.-J. KAZEMZADEH-PARSI, A. AMMAR, D. BAILLARGEAT, AND F. CHINESTA, *The importance of wind simulation to find in real time the optimal flight trajectory in drones operations*, 10th OpenFOAM Conference, November 8 2022
- 8. V. CHAMPANEY, D. DI LORENZO, A. PASQUALE, A. AMMAR, M.-J. KAZEMZADEH-PARSI, D. BAILLARGEAT, E. CUETO, AND F. CHINESTA, *Hybrid modelling for decision making in critical systems*, IUTAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022
- 9. A. RUNACHER, A. PASQUALE, M.-J. KAZEMZADEH-PARSI, C. JOFFRE, C. SALVAN, C. GHNATIOS, E. CUETO, AND F. CHINESTA, *Data-driven modelling of the filament shape evolution in fused filament fabrication process*, IU-TAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022
- 10. M. JACOT, V. CHAMPANEY, A. PASQUALE, P. TRIGUERO NAVARRO, AND F. CHINESTA, Real-time structural health monitoring of aeronautics structures: a damage detection and identification approach based on a multidimensional parametric model using the sparse proper generalized decomposition coupled with optimization, IUTAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022
- A. PASQUALE, V. CHAMPANEY, A. AMMAR, F. CHINESTA, AND J.-L. DUVAL, *Pgd based time multiscale*, Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering and Technology (MMLDT-CSET), Septem-ber 26-29 2021

2 Space

Abstract_

Plate and shell structures are commonly found in many mechanical systems. In this context, the 2D simulation based on the shell theory appears as an appealing choice since it reduces the computational complexity. Nevertheless, this 2D framework may fail to capture rich physics (such as plastic flow localization, stress concentration, striction and fracture) compromising the usual assumptions made in shell theories and, in these scenarios, a 3D discretization becomes compulsory. Unfortunately, the resultant meshes often entail an excessive number of degrees of freedom, as the mesh granularity needs to proportionately scale with the thickness dimension to prevent overly distorted elements.

In this chapter, we propose a novel finite element formulation for thin to thick structures, based on enriching the first-order Reissner-Mindlin 2D theory. Such formulation preserves the in-plane kinematics of the classical first-order theory, while enriching the out-of-plane one via higher-order polynomials. This is done without increasing the number of degrees of freedom of the standard shell element, but using a machine learning (ML) model to predict the out-of-plane polynomials.

We present the implementation prototype in an industrial solver, limiting the intrusiveness of the procedure. Afterwards, we consider some benchmark tests in elasticity and elasto-plasticity where classical shell elements would lead to inaccurate mechanical predictions.

Contents

2.1 Introduction 20	
2.2 Shell theory	
2.3 Enhanced theory	
2.3.1 Finite element formulation of the in-plane problem $\ldots \ldots 26$	
2.3.2 Out-of-plane kinematics estimation $\ldots \ldots \ldots \ldots \ldots 30$	
2.4 3D simulations 31	
2.4.1 Test case 1 \ldots 31	
2.4.2 Test case 2 \ldots 32	
2.4.3 Test case 3 \ldots 34	
2.4.4 Test case 4 $\ldots \ldots \ldots \ldots \ldots \ldots 36$	
2.4.5 Test case 5 \ldots 38	
2.5 Machine learning based training	

2.1 INTRODUCTION

Most of mechanical systems and complex structures exhibit plates and shells components (i.e., having planar dimensions much larger than their thicknesses), making them crucial to computational structural mechanics [8–10,97,98]. In this field, most commonly employed approaches make use of some kinematic and mechanic assumptions to reduce the three-dimensional nature of mechanical models to two-dimensional reduced models defined in the plate (or shell) middle surface. For instance, first-order theories assume that the plate thickness remains constant during deformation and that the displacement varies linearly across the plate thickness.

Plates and shells theories are widely employed in industry [99,100] since they guarantee extremely efficient numerical simulations. For instance, the Reissner–Mindlin shell theory [101,102] is commonly used to predict the deformation of thin and moderately thick shells accounting for transverse shear deformation effects¹. A common numerical drawback of such theory is known as shear locking effect, arising in case of thin-walled structures. This can be prevented via reduced integration, among other possibilities [9].

More complex drawbacks are physical rather than numerical. Indeed, due to their inability to account for three-dimensional effects like plastic flow localization, stress concentration, striction and fracture, classical first-order theories may not always ensure reliable simulations [103]. This issue is typically encountered in sheet metal forming [104] or in multilayered composite manufacturing processes [105, 106]. Such forming processes may induce an important through-the-thickness evolution of the thermo-mechanical fields, producing residual stresses and distorted formed parts.

In these scenarios, three-dimensional models become compulsory. Nevertheless, given the degenerate nature of the geometry, the usage of meshes based on solid elements is limited. Indeed, to avoid overly distorted elements which would compromise the simulation results, the granularity of the mesh should scale with the domain thickness. This produces an extremely high number of elements resulting in a prohibitive computational cost.

This issue is faced, for instance, when employing the shell-to-solid remeshing (SSR) procedure, available in ESI VPS [107]. Here, according to specific criteria, when the shell-based simulation does not guarantee reliable results, the part of interest is extruded and a solid-based remeshing is performed. This usually involves a high number of solid elements, which significantly increases the computing time. Moreover, this procedure also entails other difficulties such as the coupling of the extruded part with the surrounding 1D beam elements or 2D shell elements, which is usually not trivial. This issues are even acerbated in case of junctions, were the remeshing inevitably causes highly skewed element ratios in the corners, as shown in figure 2.1.

Several representations and new finite elements types have been developed to avoid the just referred issues. For instance, solid-shell elements lie halfway between thinshell and traditional solid elements and are specifically designed to overcome shear locking [108–114]. As a main disadvantage, many solid-shell approaches do not have

¹Contrarily from Kirchhoff–Love plates theory, where both transverse shear and transverse normal strain effects are neglected, limiting the deformation to bending.



FIGURE 2.1: Shell-to-solid remeshing (courtesy of ESI Group).

a great meshing flexibility and opportune pre-processing steps are often required. Furthermore, although the computing time is shorter than simulations based on solid elements, it still cannot match that of shells-based ones.

Other techniques are based on enriching the shell-based descriptions to account for more complex out-of-plane behaviors, enabling thickness-stretching effects and 3D constitutive relations [115–117]. Higher-order theories enhance the kinematic assumption of classical theories considering higher-order polynomial displacements (such as quadratical or cubic) over the thickness [118, 119]. Hierarchical shell formulations suggest a parametrization of the through-the-thickness behavior via hierarchical high order functions [120].

Many advancements in this research line rely also on the Carrera unified formulation (CUF) [121–124], which arbitrarily expands unknown variables, such as displacement or stress components, to degenerate any structural theories into generalized kinematics.

Substantially different approaches are instead based on the in-plane/out-of-plane separated representations developed in the framework of the proper generalized decomposition (PGD) $[36-39]^2$. In this context, 3D solutions are computed alternating series of 2D (in-plane) and 1D (out-of-plane) problems, preserving the computational cost of a 2D simulation, while enabling a detailed and accurate representation of the solution evolution along the thickness coordinate. However, despite the effectiveness of such methods, their applicability to large-scale problems is often limited by their intrusive nature. Many attempts have been conducted in this direction to propose less intrusive PGD-based solutions [40-42, 51, 53].

The primary objective of the current study is to suggest and investigate novel options with the dual aim of improving the quality of conventional shell-based simulations when the 2D theory's predictions are questioned while maintaining the same computing cost. Many features from the works described before are inherited for this aim. The formulation specifically retains the in-plane kinematics of the traditional firstorder theory while enhancing the out-of-plane one using higher-order polynomials. As a novelty, a machine learning framework is introduced to predict the proper kinematic,

²These approaches are not limited to structural mechanics, and have been successfully applied also to other contexts, such as fluid-dynamics in narrow gaps [43–45] or electromagnetism in laminates [49, 50].

in each in-plane position, in order to avoid increasing the number of degrees of freedom of the typical shell element.

In past works, a similar technique has been proposed for the solution of parametric heat conduction problems in thin structures [125]. This work can be seen as an extension of [125] to structural mechanics.

The chapter is structured as follows. In section 2.2 the first-order shell theory is described, emphasizing its limitations. In section 2.3 the proposed enhanced theory is presented, paying attention to the implementation details in a commercial solver. In section 2.4 some numerical examples are discussed. Finally, section 2.6 gives a short conclusion and perspectives.

2.2 Shell theory

In what follows, a geometrically degenerated system is referred as a plate or shell-like geometry, representative of a generic structure having the out-of-plane dimension h (thickness) much lower than the in-plane ones. This is represented in Fig. 2.2.



FIGURE 2.2: Example of a shell-like structure.

The numerical simulation of problems defined in such domains mostly involves plate and shell elements, based on specific mechanical theories, as widely discussed in [9,10]. In particular, we will focus on Reissner–Mindlin plate theory, which is an extension of Kirchhoff plate theory. While the former works also for thick plates, the latter models only thin plate situations (thickness/average size ≤ 0.1). The classic thin plate Kirchhoff theory establishes that the normal remains straight and orthogonal to the middle plane after deformation. The more advanced thick plate theory proposed by Reissner and Mindlin assumes that normals remain straight, though not necessarily orthogonal to the middle plane after deformation. The two theories will be summerized here below.

THIN PLATES: KIRCHHOFF THEORY A thin plate is a domain $\Omega = \Omega_{xy} \times \Omega_z$ with $\Omega_z = \left[-\frac{h}{2}, \frac{h}{2}\right]$. The theory starts by reducing the structure to its middle plane (or middle surface) Ω_m corresponding to z = 0, which is also referred as the reference plane. Let $u_Q = (u, v, w)$ denote the displacement vector of a generic point x_Q belonging to the plate.

The assumptions of the Kirchhoff thin plate theory are reported here below.

1. The points belonging to the middle plane (z = 0) only move vertically, that is

$$\boldsymbol{u}_P = (0,0,w) \quad \forall \boldsymbol{x}_P \in \Omega_m.$$

- 2. The points along a normal to the middle plane have the same vertical displacement. This is equivalent to ask for the thickness of the plate to be invariant during deformation.
- 3. The normal stress σ_z is negligible, which is known as plane stress assumption.
- 4. A straight line normal to the undeformed middle plane **remains straight** and **normal** to the deformed middle plane, which is the normal orthogonality condition.

THICK/THIN PLATES: REISSNER-MINDLIN THEORY The Reissner-Mindlin plate bending theory shares the first three assumptions of Kirchhoff plate theory. However, the fourth hypothesis concerning the normal is reformulated as follows.

4. A straight line normal to the undeformed middle plane **remains straight** but **not necessarily orthogonal** to the middle plane after deformation.

DISPLACEMENT For both theories, the 3D displacement field is expressed in terms of the middle plane kinematic variables w, θ_x and θ_y :

$$\begin{cases} u(x, y, z) = z\theta_y(x, y) \\ v(x, y, z) = -z\theta_x(x, y) \\ w(x, y, z) = w(x, y) \end{cases}$$
(2.2.1)

where θ_x and θ_y are the angles defining the rotation of the normal vector to the middle surface Ω_m . Consequently, in both cases, the displacement vector contains the vertical displacement (deflection) of the points on the middle plane and two rotations:

$$\boldsymbol{u} = (\theta_y, \theta_x, w)^T. \tag{2.2.2}$$

In particular, for the Kirchhoff theory, the rotations of the normal coincide with the slopes of the middle plane at each point; this means that, in the plane xz, $\theta_x = \frac{\partial w}{\partial y}$ and, in the plane yz, $\theta_y = -\frac{\partial w}{\partial x}$. For the Reissner–Mindlin theory, the rotations of the normal are obtained with an additional rotation resulting from the lack of orthogonality of the normal with the middle plane after deformation; this means that, in the plane xz, $\theta_x = \frac{\partial w}{\partial y} + \phi_x$ and, in the plane yz, $\theta_y = -\frac{\partial w}{\partial x} + \phi_y$.

Actually, the hypothesis of straight normals in both theories is an approximation of the true plate kinematics. Indeed, in reality, the plate normals are distorted during deformation. This effect is more important for thick plates. The angles θ_x and θ_y can be interpreted as the rotations of the straight line representing the "average" deformation of the normal [9].

To summarize the plate theories, (a) the straight sections, i.e. the sections perpendicular to the mid-surface, remain straight; (b) the material points located on a normal to the undeformed mid-surface remain on a straight line in the deformed configuration. The two theories are often resumed as in figure 2.3.

These assumptions imply that the displacement fields vary linearly in the thickness of the plate. This means that a generic point $\mathbf{x}_Q(x, y, z)$ belonging to the plate follows the first-order kinematics:

$$\begin{pmatrix} u_Q(x,y,z)\\ v_Q(x,y,z)\\ w_Q(x,y,z) \end{pmatrix} = \begin{pmatrix} u_P(x,y)\\ v_P(x,y)\\ w_P(x,y) \end{pmatrix} + z \begin{pmatrix} \theta_y(x,y)\\ -\theta_x(x,y)\\ 0 \end{pmatrix}$$
(2.2.3)



FIGURE 2.3: Kirchhoff versus Reissner–Mindlin theory.

where u_P , v_P and w_P are the displacements of the middle plane and θ_x , θ_y the rotations of this surface with respect to the two axis x and y, respectively. For what concerns shear strains γ_{xz} and γ_{yz} , they are null in the Kirchhoff thin plate theory while constant through-the-thickness in the Reissner-Mindlin thick plate theory.

A shell-elements based simulation can be viewed as represented in figure 2.4. In particular, the degenerated solid Ω of figure 2.2 is reduced to its middle surface Ω_m , which is discretized in surface elements A^e , having five degrees of freedom per node $\boldsymbol{u}_i^e = (u_i, v_i, w_i, \theta_x^i, \theta_y^i)$. Moreover, a given number of integration points is fixed through-the-thickness (in the figure three points). The straight lines illustrate the through-the-thickness kinematics.



FIGURE 2.4: Illustration of a 2D shell-elements based simulation.

In terms of stresses, the thick plate theory recovers the shear stresses τ_{xz} and τ_{yz} which is more consistent than the thin plate one. However, both theories consider $\sigma_{zz} = 0$, which does not happen in the exact three-dimensional elasticity.

However, when curvatures are no longer small and shear strains become important, the main approximation of straight normals after deformation become doubtful and a fully 3D finite-elements based simulation has to be carried out. A 3D-based simulation is schematized in figure 2.5. One can observe that here solid elements having three degrees of freedom are considered. The computed kinematics can be of higher order and can vary according to the in-plane location. This implies that 3D effects such as striction can be observed (ie. h' < h).

However, a fully 3D discretization of such domains, discussed in the introductory section, would imply too distorted elements or a unaffordable huge number of regular elements, due to the extremely small thickness.



FIGURE 2.5: Illustration of a 3D finite-elements based simulation.

2.3 ENHANCED THEORY

Equations (2.2.3) mean that the displacement in the thin/thick plate is a single-mode separated approximation of in-plane and out-of-plane functions. Actually the out-of-plane contribution is uniform across the plate in the sense that it does not depend on the in-plane nodal position. In each in-plane location, displacements u and v are linear across the thickness, while the in-plane contribution is given by the rotations of the mid-surface. The out-of-plane displacement w is instead constant along the thickness.

The linear variation of the displacement field along the thickness is not a valid hypothesis when the domain the thickness (out-of-plane) dimension is not much lower than the other ones (in-plane) dimensions, for instance. Hypotheses are invalid also in case of plastic behaviours or other kind of localized deformations.

In [40] authors extend the classical PGD-based in-plane/out-of-plane separated representation in structural mechanics. In such work, authors show satisfactory results in elasticity and plasticity, employing a multi-mode separated approximation, which reads

$$\boldsymbol{u}(x,y,z) = \begin{pmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{pmatrix} \approx \sum_{i=1}^{N} \begin{pmatrix} \theta_{y}^{i}(x,y) f_{u}^{i}(z) \\ \theta_{x}^{i}(x,y) f_{v}^{i}(z) \\ w^{i}(x,y) f_{w}^{i}(z) \end{pmatrix}.$$
 (2.3.1)

This is actually a way to avoid simplification hypothesis related to plate theories, and to capture a richer behaviour along the thickness. Indeed the PGD solution is comparable to a fully 3D FEM simulation. For instance, in [40], it is shown how the proposed method is able to take into consideration the σ_{zz} component, which is ignored in plate theory, and allows to obtain the parabolic evolution around the thickness for the σ_{xy} and σ_{yz} typical of a 3D solution. At the meanwhile, its intrusiveness limits the implementation of such procedure into commercial software.

In this thesis, the theory is expressed in a less intrusive way to favor the implementation within ESI VPS software. To this purpose, we consider an enriched formulation of the plate theory, based on allowing a higher order kinematics, through higher-order out-of-plane functions, supposed to be known. The enriched-theory displacement field of a generic point $\mathbf{x}_Q(x, y, z)$ reads

$$\begin{pmatrix} u_Q(x,y,z)\\ v_Q(x,y,z)\\ w_Q(x,y,z) \end{pmatrix} = \begin{pmatrix} u_P(x,y)\\ v_P(x,y)\\ w_P(x,y) \end{pmatrix} + \begin{pmatrix} g_x(z)\theta_y(x,y)\\ g_y(z)\theta_x(x,y)\\ g_z(z)w(x,y) \end{pmatrix}$$
(2.3.2)

which coincides with the first order theory (2.2.3) for $g_x(z) = z, g_y(z) = -z, g_z(z) = 0$. The geometry is still based on the planer facet approach (first order):

$$oldsymbol{x}_Q = oldsymbol{x}_P + zoldsymbol{n}, \quad z \in igg[-rac{h}{2},rac{h}{2}igg].$$

The through-the-thickness functions $g_x(z), g_y(z), g_z(z)$ are generic polynomial assumed known and depending only on the z direction. Within the enriched theory θ_x, θ_y, w do not represent rotations and deflection as those of Reissner-Mindlin theory explained before.

Afterwards, the higher-order-kinematics can be embedded within the 2D computation of the in-plane functions. For this purpose the 3D mesh is associated to the 2D one obtained through its middle surface. The general procedure is based on defining in the undeformed configuration columns C^e of 3D elements which are associated to the shell mesh via the middle element A_e . This is schematized in figure 2.6.



FIGURE 2.6: Association of solid-based mesh to shell-based one.

The implemented flowchart is illustrated in figure 2.7. Starting from the assumed displacement (2.3.2), an enriched problem can be assembled within the plane. The outof-plane kinematic functions are predicted by a machine learning framework which is trained exploiting results coming from a 3D (parametric or non-parametric) simulation (which can be PGD-based [40] if the FE one is not affordable computationally).

2.3.1 FINITE ELEMENT FORMULATION OF THE IN-PLANE PROBLEM

The finite element approximation starts by introducing a partition of the domain Ω_m into non-overlapping subdomains (or, finite elements) A^e as $\Omega_m \approx \bigcup_{e=1}^N A^e$, as illustrated in figure 2.6 (right). Same occurs for the boundary, which is decomposed as $\Gamma_m \approx \bigcup_{e=1}^N \Gamma^e = \bigcup_{e=1}^N \Gamma^e_d \cup \bigcup_{e=1}^N \Gamma^e_t$. We have denoted by Γ^e_d and by Γ^e_t the finite boundary elements with enforced displacement and traction, respectively. The displacement field associated with a local element reads

$$\boldsymbol{u}(\boldsymbol{x},t) \approx \sum_{i=1}^{N_{\text{loc}}} N_i(\boldsymbol{x}) \boldsymbol{u}_i^e(t) = \boldsymbol{N}^T(\boldsymbol{x}) \boldsymbol{u}^e(t)$$

where $N_i(\boldsymbol{x})$ are element shape functions, while $\boldsymbol{u}_i^e(t)$ are the time-dependent nodal displacements and N_{loc} denotes the number of degrees of freedom of the element A^e .



FIGURE 2.7: Implementation prototype in ESI VPS – Creation of a dynamic library containing a machine learning model framework.

In what follows, we consider a four-node and bilinear two-dimensional quadrilateral element, with nodal variables $\boldsymbol{u}_i^e = (u_i, v_i, w_i, \theta_x^i, \theta_y^i)$, for $i = 1, \ldots, 4$.

In isoparametric form, the finite element approximation of the displacement field is

$$egin{aligned} oldsymbol{u}(oldsymbol{\xi},t) &pprox \sum_{i=1}^{N_{ ext{loc}}} N_i(oldsymbol{\xi})oldsymbol{u}^e_i(t) = oldsymbol{N}^T(oldsymbol{\xi})oldsymbol{u}^e(t) \ oldsymbol{x}(oldsymbol{\xi}) &= \sum_{i=1}^{N_{ ext{loc}}} N_i(oldsymbol{\xi})oldsymbol{x}^e_i = oldsymbol{N}^T(oldsymbol{\xi})oldsymbol{x}^e \end{aligned}$$

where x^e collects the nodal coordinates of the element A^e in the physical space, while $\boldsymbol{\xi}$ are the parametric (reference) coordinates. The derivative of shape functions with respect to parametric coordinates are obtained by the classical chain rule, accounting for the Jacobian \boldsymbol{J} of the transformation between $\boldsymbol{\xi}$ and \boldsymbol{x} :

$$N_{i,\boldsymbol{\xi}} = \boldsymbol{J} N_{i,\boldsymbol{x}}, \quad N_{i,\boldsymbol{\xi}_m} = x_{l,\boldsymbol{\xi}_m} N_{i,x_l}.$$

Consequently, the derivative of the shape functions with respect to the physical coordinate is $N_{i,x} = J^{-1}N_{i,\xi}$.

EQUATION OF MOTION The equation of motion reads:

$$\boldsymbol{f}_v - \rho \boldsymbol{\ddot{u}} - C \boldsymbol{\dot{u}} - \nabla \cdot \boldsymbol{\sigma} = \boldsymbol{0}$$

where f_v is the force vector per unit volume, ρ is the mass density, C is the damping coefficient and σ the stress tensor.

Following the virtual power principle, we multiply the equation of motion by the virtual velocity function $\dot{\boldsymbol{u}}^*$, integrate over the current configuration A^e and apply the divergence theorem. This leads to

$$\boldsymbol{M}^{e} \ddot{\boldsymbol{u}}^{e} + \boldsymbol{C}^{e} \dot{\boldsymbol{u}}^{e} + \boldsymbol{f}_{\text{int}}^{e} = \boldsymbol{f}_{\text{ext}}^{e}$$
(2.3.3)

where

- $M^e = \int_{A^e} \rho N^T N dA^e$ is the element mass matrix;
- $C^e = \int_{A^e} C N^T N dA^e$ is the element damping matrix;
- $f_{\text{int}}^e = \int_{A^e} B^T \sigma dA^e$ is the internal force vector;
- $\boldsymbol{f}_{\text{ext}}^{e} = \int_{A^{e}} \boldsymbol{N}^{T} \boldsymbol{f}_{v} dA^{e} + \int_{\Gamma^{e}} \boldsymbol{N}^{T} \boldsymbol{f}_{s} d\Gamma^{e} + \int_{\Gamma^{e}} \boldsymbol{N}^{T} \boldsymbol{r} d\Gamma^{e}$ is the external force vector (volume, surface and source loads);
- \ddot{u}^e and \dot{u}^e are the vectors of nodal acceleration and velocity;
- N(ξ, η) the Lagrange shape functions and B the strain operator such that ε^e = Bu^e.

To compute the strain operator \boldsymbol{B} of the higher-order element formulation, we consider equation 2.3.2 and we develop the displacement field and the strain field. For the sake of notation simplicity the index related to the current element is omitted.

DISPLACEMENT The displacement field is expanded as

$$u_Q(x, y, z) = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) u_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) u_P^i + g_x(z) \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) \theta_y^i$$
$$v_Q(x, y, z) = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) v_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) v_P^i + g_y(z) \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) \theta_x^i$$
$$w_Q(x, y, z) = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_P^i + g_z(z) \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_i.$$

STRAIN The corresponding strain is obtained as follows

$$\begin{split} \varepsilon_{xx}(x,y,z) &= u_{Q,x} = u_{P,x} + g_x(z)\theta_{y,x} \\ &= \sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)u_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)u_P^i + g_x(z)\sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)\theta_y^i \\ \varepsilon_{yy}(x,y,z) &= v_{Q,y} = v_{P,y} + g_y(z)\theta_{x,y} \\ &= \sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi,\eta)v_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi,\eta)v_P^i + g_y(z)\sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi,\eta)\theta_x^i \\ \gamma_{xy}(x,y,z) &= u_{Q,y} + v_{Q,x} = u_{P,y} + v_{P,x} + g_x(z)\theta_{y,y} + g_y(z)\theta_{x,x} \\ &= \sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi,\eta)u_Q^i + \sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)v_Q^i \\ &+ g_x(z)\sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi,\eta)\theta_y^i + g_y(z)\sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)\theta_x^i \\ \gamma_{xz}(x,y,z) &= u_{Q,z} + w_{Q,x} = g_{x,z}(z)\theta_y + w_{P,x} + g_z(z)w_{,x} \\ &= g_{x,z}\sum_{i=1}^{N_{\text{loc}}} N_i(\xi,\eta)\theta_y^i + (1+g_z(z))\sum_{i=1}^{N_{\text{loc}}} N_{i,x}(\xi,\eta)w_i \\ \gamma_{yz}(x,y,z) &= v_{Q,z} + w_{Q,y} = g_{y,z}(z)\theta_x + w_{P,y} + g_z(z)w_{,y} \end{split}$$

$$= g_{y,z} \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) \theta_x^i + (1 + g_z(z)) \sum_{i=1}^{N_{\text{loc}}} N_{i,y}(\xi, \eta) w_i$$
$$\varepsilon_{zz}(x, y, z) = w_{Q,z} + g_{z,z}(z) w = g_{z,z} \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_i.$$

We can then express the strain operator for the enriched formulation as

$$egin{aligned} oldsymbol{arepsilon}^e &= oldsymbol{B}^e oldsymbol{u}^e \ oldsymbol{arepsilon}^e &= egin{pmatrix} oldsymbol{u}_1 & oldsymbol{B}_2 & oldsymbol{B}_3 & oldsymbol{B}_4 \end{pmatrix} egin{pmatrix} oldsymbol{u}_1 \ oldsymbol{u}_2 \ oldsymbol{u}_3 \ oldsymbol{u}_3 \ oldsymbol{u}_4 \end{pmatrix}, \end{aligned}$$

where $\boldsymbol{u}_i = (u_i, v_i, w_i, \theta_x^i, \theta_y^i)$, for i = 1, ..., 4. In particular, the table here below illustrates the application of the operator B_i to the displacement vector \boldsymbol{u}_i .

	u_i	v_i	w_i	$ heta_x^i$	$ heta_y^i$
ε_x	$N_{i,x}$				$g_x(z)N_{i,x}$
ε_y		$N_{i,y}$		$g_y(z)N_{i,y}$	
ε_z			$g_{z,z}(z)N_i$		
γ_{xy}	$N_{i,y}$	$N_{i,x}$		$g_y(z)N_{i,x}$	$g_x(z)N_{i,y}$
γ_{xz}			$(1+g_z(z))N_{i,x}$		$g_{x,z}(z)N_i$
γ_{yz}			$(1+g_z(z))N_{i,y}$	$g_{y,z}(z)N_i$	

TABLE 2.1: Operator B_i , for $i = 1, \ldots, 4$.

From these computations, one can notice that the local assembly of problem 2.3.3 requires the integration of quantities over A^e , which can be reduced to separated integrals over A_e for the in-plane functions and over $\left[-\frac{h}{2}, \frac{h}{2}\right]$ for the out-of-plane functions. Indeed, the assembly of the internal force vector can be developed as follows

$$\boldsymbol{f}_{\text{int}}^{e} = \int_{A^{e}} \boldsymbol{B}^{T} \boldsymbol{\sigma} dA^{e} = \int_{-\frac{h}{2}}^{\frac{h}{2}} \int_{A_{e}} \boldsymbol{B}^{T} \boldsymbol{\sigma} dA_{e} dz$$

and, considering the computations in Table 2.1, we can explicitly write the force and momentum assembly of each in-plane mesh node i:

$$\begin{split} f_x &= \int_{A_e} N_{i,x} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{xx} \mathrm{d}z + \int_{A_e} N_{i,y} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{xy} \mathrm{d}z \\ f_y &= \int_{A_e} N_{i,y} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{yy} \mathrm{d}z + \int_{A_e} N_{i,x} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} \sigma_{xy} \mathrm{d}z \\ f_z &= \int_{A_e} N_i \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_{z,z}(z) \sigma_{zz} \mathrm{d}z + \int_{A_e} N_{i,x} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} (1 + g_z(z)) \sigma_{xz} \mathrm{d}z + \\ \int_{A_e} N_{i,y} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} (1 + g_z(z)) \sigma_{yz} \mathrm{d}z \\ m_x &= \int_{A_e} N_{i,y} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_y(z) \sigma_{yy} \mathrm{d}z + \int_{A_e} N_{i,x} \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_y(z) \sigma_{xy} \mathrm{d}z + \end{split}$$

$$\int_{A_e} N_i \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_{y,z}(z)\sigma_{yz}\mathrm{d}z$$

$$n_y = \int_{A_e} N_{i,x}\mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_x(z)\sigma_{xx}\mathrm{d}z + \int_{A_e} N_{i,y}\mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_x(z)\sigma_{xy}\mathrm{d}z + \int_{A_e} N_i \mathrm{d}A^e \int_{-\frac{h}{2}}^{\frac{h}{2}} g_{x,z}(z)\sigma_{xz}\mathrm{d}z.$$

The computation of the in-plane and out-of-plane integrals is carried out through standard Gaussian quadrature.

2.3.2 OUT-OF-PLANE KINEMATICS ESTIMATION

At this point only functions $g_x(z), g_y(z), g_z(z)$ are needed to finalize of the operators and right-hand-side via computation of all the out-of-plane integrals.

The enhanced kinematics can be extracted from 3D simulations performed in an initial offline stage. The through-the-thickness displacements of C^e are reconstructed and fitted with higher-order polynomial functions associated to the corresponding shell element A^e . The out-of-plane functions can be provided per element or per node, at each time step or could be unchanged over a given time interval. Supposing that each node has an associated out-of-plane function, for instance, the z-component of the displacement over an element is expanded as

$$w_Q(x, y, z) = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_Q^i = \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) w_P^i + \sum_{i=1}^{N_{\text{loc}}} N_i(\xi, \eta) g_z^i(z) w_i,$$

meaning that $g_z^i(z)$ acts like a local weight allowing a richer out-of-plane physics.

Afterwards, all the integrals can be computed and the in-plane problem solved. In such a way, the higher-order-kinematics can be embedded within the 2D computation of the in-plane functions. The procedure is schematized in figure 2.8.



FIGURE 2.8: Illustration of a 2D enhanced simulation.

1

2.4 3D SIMULATIONS

In this section, we give a few numerical examples in elasticity and elasto-plasticity. When plasticity is considered, the chosen model is a standard von Mises (J_2) plasticity, whose yield surface is defined by

$$\Phi(\boldsymbol{\sigma}(t), \boldsymbol{\varepsilon}^{p}(t)) = \sqrt{3J_2} - \sigma_{y,t} = 0, \qquad (2.4.1)$$

where $J_2 = J_2(t)$ denotes the second deviatoric invariant

$$J_2 = J_2(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s} : \boldsymbol{s}, \quad \boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\sigma})\boldsymbol{I}$$
(2.4.2)

and $\sigma_{y,t}$ is the uniaxial yield stress, which evolves through a suitable strain-hardening curve. The details of this evaluation are given below.

Making the assumption of isotropic hardening, at any state of hardening, the evolution of the yield surface (3.2.16) corresponds to a uniform (isotropic) expansion of the initial one. This is obtained assuming $\sigma_{y,t}$ being a function of the accumulated (or effective) plastic strain

$$\bar{\varepsilon}_t^p = \int_0^t \sqrt{\frac{2}{3}} \dot{\varepsilon}^p : \dot{\varepsilon}^p \mathrm{d}s.$$
 (2.4.3)

In particular, assuming linear hardening, $\sigma_{y,t}$ is given by

$$\sigma_{y,t} = \sigma_{y,0} + H\bar{\varepsilon}_t^p, \qquad (2.4.4)$$

where $\sigma_{y,0}$ is the initial yield stress and H is the (constant) hardening modulus.

Additionally, a standard associative plastic flow rule is considered, meaning that the plastic strain rate is a tensor normal to the yield surface in the stress space, that is

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda} \boldsymbol{N}, \quad \boldsymbol{N} := \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}.$$
 (2.4.5)

Following usual notations, in equation (3.2.20), λ denotes the unknown plastic multiplier. Specifically, when considering an associative hardening rule, it holds [126]

$$\dot{\bar{\varepsilon}}^p = \sqrt{\frac{2}{3}} \|\dot{\varepsilon}^p\| = \dot{\lambda}. \tag{2.4.6}$$

For every test case, some in-plane locations are considered (indicated over the mesh with red dots), to illustrate the out-of-plane evolution of specific quantities of interest (such as the displacement and stress components).

2.4.1 TEST CASE 1

Here a clamped square thick elastic plate under distributed loading is considered. The simulation set-up is illustrated in figure 2.9 and the simulation parameters reported in table 2.9.

l	h	E	ν	F	N_{xy}	N_z
500 [mm]	100 [mm]	195 · 10 ⁹ [GPa]	0.3	200 [kN]	400	10

TABLE 2.2: Simulation parameters (test case 1).



FIGURE 2.9: Geometry and loading conditions (test case 1).



FIGURE 2.10: 3D displacement contours (test case 1).

The contour plot of the displacement components is given in figure 2.10.

In particular, we can focus on the out-of-plane displacement and stress profiles, shown in figures 2.11 and 2.12, respectively. The red points illustrated over the structure represent the in-plane locations in which the out-of-plane evolution is considered.

This example shows out-of-plane displacements with cubic evolution in components u and v and a quadratic profile for the component w. Moreover, from figure 2.12 we can observe non-negligible shear stresses, as would instead be predicted by the first-order theory.



FIGURE 2.11: Out-of-plane displacement profiles (test case 1).

2.4.2 TEST CASE 2

Here a clamped square thick elastic plate under compressive loading is considered. Such kind of compression is crucial for consolidating the layers and removing any trapped air or excess resin in composite laminates manufacturing process. A proper consolidation improves the mechanical properties of the composite.

The simulation set-up is illustrated in figure 2.13 and the simulation parameters reported in table 2.13.



FIGURE 2.12: Out-of-plane stress profiles (test case 1).



FIGURE 2.13: Geometry and loading conditions (test case 2).

l	h	E	ν	w_0	N_{xy}	N_z
500 [mm]	100 [mm]	$195 \cdot 10^9 [\text{GPa}]$	0.3	10 [mm]	400	10

TABLE 2.3: Simulation parameters (test case 2).

The contour plot of the displacement components is given in figure 2.14.



FIGURE 2.14: 3D displacement contours (test case 2).

The out-of-plane displacement and stress profiles are shown in figures 2.15 and 2.16, respectively. The red points illustrated over the structure represent the in-plane locations in which the out-of-plane evolution is considered.

This example shows out-of-plane displacements with quadratic evolution in components u and v and a linear profile (constant in the plane) for the component w. This example could not be treated using the first-order theory since it implies a change of thickness. Moreover, in terms of stresses, polynomial profiles are computed. The evolution is quadratic for the normal stresses σ_{xx}, σ_{yy} and σ_{zz} and for the in-plane shear stress σ_{xy} . The out-of-plane shear stresses σ_{xz} and σ_{yz} have a cubic evolution.



FIGURE 2.15: Out-of-plane displacement profiles (test case 2).



FIGURE 2.16: Out-of-plane stress profiles (test case 2).

2.4.3 TEST CASE 3

Here an elastic-plastic thick plate under 3-points bending test is considered. The simulation set-up is illustrated in figure 2.17 and the simulation parameters are fixed according table 2.4.



FIGURE 2.17: Geometry and loading conditions (test case 3).

L_x	L_y	h	E	ν	w_0	$\sigma_{y,0}$	H	N _{xy}	N_z
10 [cm]	6 [cm]	1 [cm]	195 [GPa]	0.3	4 [mm]	502 [MPa]	0.05 [GPa]	400	10

TABLE 2.4: Simulation parameters (test case 3).

The computed displacement contours are shown in figure 2.18. Moreover, figure 2.19 furnishes the contour of the plastic strain and Von Mises stress.



FIGURE 2.18: 3D displacement contours (test case 3).



FIGURE 2.19: Plastic strain and Von Mises stress contours (test case 3).

In particular, figure 2.20 gives the effective plastic strain and plastic strain tensor components out-of-plane profiles, corresponding to the in-plane locations exhibiting maximum effective plastic strain. As one can observe from these profiles, for z between 0.002 and 0.004 the structure is in elastic regimen.



FIGURE 2.20: Out-of-plane plastic strain profiles (test case 3).

The corresponding displacements and stresses out-of-plane profiles are shown in figures 2.21 and 2.22, respectively. As expected from a 3 points bending test, the component v is null (aside from some boundary effects). The components u and w are linear and quadratic, respectively.



FIGURE 2.21: Out-of-plane displacement profiles (test case 3).



FIGURE 2.22: Out-of-plane stress profiles (test case 3).

2.4.4 TEST CASE 4

Here a T-joint specimen under flexion is considered, as illustrated in the test set-up of figure 2.23. The simulation parameters are fixed according table 2.5.



FIGURE 2.23: Geometry and loading conditions (test case 4).

l_1	l_2	h	E	ν	w_0	$\sigma_{y,0}$	Н	N _{xy}	N_z
10 [cm]	3 [cm]	$1.5 [{\rm cm}]$	195 [GPa]	0.3	4 [mm]	502 [MPa]	0.05 [GPa]	114	6

TABLE 2.5: Simulation parameters (test case 4).

The computed displacement contours are shown in figure 2.24. Moreover, figure 2.25 furnishes the contour of the plastic strain and Von Mises stress.

In particular, figure 2.26 gives the plastic strain out-of-plane profiles, corresponding to the in-plane locations exhibiting maximum effective plastic strain. As one can observe from these profiles, for z between 0.005 and 0.01 the structure is in elastic regimen.

The corresponding displacements and stresses out-of-plane profiles are shown in figures 2.27 and 2.28, respectively. In this use-case, we can observe that the first order theory well describes the solution and the higher-order enhancement is not necessary.



FIGURE 2.24: 3D displacement contours (test case 4).



FIGURE 2.25: Plastic strain and Von Mises stress contours (test case 4).



FIGURE 2.26: Out-of-plane plastic strain profiles (test case 4).



FIGURE 2.27: Out-of-plane displacement profiles (test case 4).



FIGURE 2.28: Out-of-plane stress profiles (test case 4).

2.4.5 TEST CASE 5

Here a tensile test over a dogbone shaped specimen is considered, as illustrated in the test set-up of figure 2.29. The simulation parameters are fixed according table 2.6.



FIGURE 2.29: Geometry and loading conditions (test case 5).

L_x	L_y	h	E	ν	u_0	$\sigma_{y,0}$	Н	N_{xy}	N_z
10 [cm]	$2.5 [\rm{cm}]$	1 [cm]	195 [GPa]	0.3	8 [mm]	502 [MPa]	0.05 [GPa]	500	10

TABLE 2.6: Simulation parameters (test case 5).

The computed displacement contours are shown in figure 2.30. Moreover, figure 2.31 furnishes the contour of the plastic strain and Von Mises stress.



FIGURE 2.30: 3D displacement contours (test case 5).

In particular, figure 2.32 gives the plastic strain out-of-plane profiles, corresponding to the in-plane locations exhibiting maximum effective plastic strain.

The corresponding displacements and stresses out-of-plane profiles are shown in figures 2.33 and 2.34, respectively. We can observe that the kinematics differs from the one assumed by the first order theory and the enhanced theory is necessary to obtain three-dimensional effects.



FIGURE 2.31: Plastic strain and Von Mises stress contours (test case 5).



FIGURE 2.32: Out-of-plane plastic strain profiles (test case 5).



FIGURE 2.33: Out-of-plane displacement profiles (test case 5).



FIGURE 2.34: Out-of-plane stress profiles (test case 5).

2.5 MACHINE LEARNING BASED TRAINING

For a given geometric configuration and fixed material properties (e.g., for each test case illustrated in subsection 2.4), from the high-fidelity 3D simulation, in each in-

plane location $i = 1, ..., N_{xy}$, we can approximate the out-of-plane displacements with a polynomial function. For instance, considering a third order polynomial, the approximation reads

$$\begin{cases} u_i(z) \approx u_i^p(z) = b_0^{u,i} + b_1^{u,i}z + b_2^{u,i}z^2 + b_3^{w,i}z^3\\ v_i(z) \approx v_i^p(z) = b_0^{v,i} + b_1^{v,i}z + b_2^{v,i}z^2 + b_3^{w,i}z^3\\ w_i(z) \approx w_i^p(z) = b_0^{w,i} + b_1^{w,i}z + b_2^{w,i}z^2 + b_3^{w,i}z^3. \end{cases}$$

To this purpose, it suffices to fill the matrices $\mathbf{U}, \mathbf{V}, \mathbf{W} \in \mathbb{R}^{N_z \times N_{xy}}$ containing the out-of-plane values to be fitted at every in-plane location locations (by columns). The unknown polynomial coefficients collected in the matrices $\mathbf{b}_{\mathbf{U}}, \mathbf{b}_{\mathbf{V}}, \mathbf{b}_{\mathbf{W}} \in \mathbb{R}^{4 \times N_{xy}}$ are then obtained via the least square method, by solving

$$\begin{cases} (\mathbf{Z}_3^T \mathbf{Z}_3) \mathbf{b}_{\mathbf{U}} = \mathbf{Z}_3^T \mathbf{U} \\ (\mathbf{Z}_3^T \mathbf{Z}_3) \mathbf{b}_{\mathbf{V}} = \mathbf{Z}_3^T \mathbf{V} \\ (\mathbf{Z}_3^T \mathbf{Z}_3) \mathbf{b}_{\mathbf{W}} = \mathbf{Z}_3^T \mathbf{W}, \end{cases}$$

where $\mathbf{Z}_3 = (\mathbf{1}, \mathbf{z}, \mathbf{z}^2, \mathbf{z}^3) \in \mathbb{R}^{N_z \times 4}$ is the Vandermonde matrix of order 3. For each inplane location, we have therefore computed four coefficients representing the related out-of-plane function.

At this stage, we can consider a design of experiments (DOE) based on N_s combinations of problem parameters, which can be geometrical, material and modeling properties. Denoting with p_k , for $k = 1, ..., N_s$, a given tuple of parameters, from the output of the corresponding 3D simulation we can compute the associated out-of-plane coefficients matrices, that is

$$(\mathbf{b}_{\mathbf{F}}^1,\ldots,\mathbf{b}_{\mathbf{F}}^{N_s}) \in \mathbb{R}^{4 \times (N_{xy}N_s)}, \quad \mathbf{F} = \mathbf{U}, \mathbf{V}, \mathbf{W}.$$

These data can be exploited to build a regression between the parameters and the associated polynomial coefficients, using for instance artificial neural-networks [125].

2.6 CONCLUSIONS

In this work, we have proposed a new methodology to incorporate three-dimensional effects in shell-based simulations. The proposed formulation enhances the kinematics assumed by usual first-order plates theories, introducing higher-order ones learnable via a machine learning framework.

The procedure is inspired by the in-plane/out-of-plane space separation promoted in the framework of the proper generalized decomposition. However, to favor its implementation in ESI VPS (or other commercial software), the in-plane/out-of-plane formulation is recast in a non-intrusive manner. In this sense, the out-of-plane function is learned in an offline phase, while the in-plane (kinematically-enhanced) computation is performed within the industrial solver.

Several examples in elasticity and elasto-plasticity, under different geometric and loading configurations, have been presented, to show the interest of the procedure. Current research is focusing on the architecture of machine learning framework able to learn the out-of-plane kinematics for specific industrial parts.

3 TIME

Abstract_

Many engineering problems are defined in very large time intervals (e.g., when dealing with fatigue, aging, dynamics with loadings involving multiple characteristic times) and, at the same time, the response must encompass the different time scales present in the model. Following standard time marching approaches, a suitable time step that captures the evolution of the finest scale has to be adopted to ensure a reliable modeling, leading to a prohibitive simulation cost.

In this chapter, we investigate a time marching scheme based on a tensorial decomposition of the time axis, straightforwardly introduced in the framework of the proper generalized decomposition. The time coordinate is transformed into a multi-dimensional time through new separated coordinates, the micro and the macro times. From a physical viewpoint, the time evolution of all the quantities involved in the problem can be followed along two time scales, the fast one (micro-scale) and the slow one (macro-scale).

We apply the method to compute a time multiscale response of an elastic-plastic structure under cyclic loading. Such micro-macro characterization of the time response is then exploited to build a data-driven model of the elasto-plastic constitutive relation, with the aim of addressing long time-horizon simulations. This can be viewed as a predictor-corrector scheme where the prediction is driven by the macrotime evolution and the correction is performed via a sparse sampling in space. Once the nonlinear term is forecast, the multi-time PGD algorithm allows the fast computation of the total strain.

Contents

3.1 Intr	oduction																		
3.2 Mul	ti-time PGD in cyclic elasto-plasticity 43																		
3.2.1	Theoretical framework $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 43$																		
3.2.2	Material and methods $\ldots \ldots \ldots \ldots \ldots \ldots \ldots 47$																		
3.2.3	Results and discussion $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 52$																		
3.2.4	Conclusions																		
3.3 Mul	ti-time based learning of history-dependent nonlinear																		
beha	aviors																		
3.3.1	Theoretical and numerical framework $\ldots \ldots \ldots \ldots 59$																		
3.3.2	Multiscale-based data-driven modeling																		
3.3.3	Results and discussion $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 71$																		
	3.3.4	Conclusi	ons						•					•				79	
-----	-------	----------	-----	--	--	--	--	--	---	--	--	--	---	---	--	--	---	----	--
3.4	Cond	clusions											•	•			•	80	

3.1 INTRODUCTION

Due to the dramatically long duration of the phenomenon and the stringent requirements on the grid granularity, direct simulations of structures subject to a high number of loading cycles remains a real challenge.

This issue is often encountered in computational mechanics, when dealing with cumulative fatigue damage assessments [127, 128], crack initiation and failure propagation [129] or cyclic visco-elasto-plastic fatigue problems [130–132]. For instance, standard numerical techniques fail in simulating the fatigue life, representing a major design issue in various fields of applications such as aircraft, auto parts, railways and jet engines, among many others [12, 13].

One of the reasons of the excessive complexity stands in the history-dependent behaviours which require the reconstruction of the whole past history [133–138]. Indeed, when this is combined with fine spatial meshes and very long time horizons, the computational complexity leads to cost-prohibitive simulations and to the necessity of adopting suitable simplified models.

Different works have been proposed in this direction, most of them applied to the large time increment (LATIN) method [139] together with the proper generalized decomposition (PGD). For instance, several multiscale approximations have been developed in [140–145], where the computational time is reduced via an interpolation of the solution at different time scales. In this context, global admissibility conditions and constitutive relations are imposed along the fast scale defined within the so-called "nodal cycle" and the information between the "nodal cycles" is interpolated to define the slow scale.

Other works rely on some hyper-reduction techniques, such as the reference point method [146,147] or the extension of the gappy-POD technique to the space-time domain [56], but these were restricted for nonlinear behaviors at internal variables, which is not the case for history-dependent behaviors. In particular, in [56], the concept of temporal sub-modes was introduced in LATIN-PGD approximations of nonlinear problems in solid mechanics. Here, the continuity of micro functions is ensured by means of symmetric-antisymmetric characterizations, while the macro functions are computed by using the finite element method in time.

In this thesis, we focus specifically on the multi-time separated representation proposed in the framework of the PGD [54,55,148]. Micro (fast) and macro (slow) time separated coordinates, τ and T, are defined and the corresponding evolution computed in terms of PGD modes. Such multiscale approach does not require time scale separation, contrarily to time homogenization techniques. The term separation in this work is, actually, meant in the context of separation of variables (i.e., PGD-like). The two time scales are coexisting within the formulation meaning that kinematics and mechanics variables are computed simultaneously along the micro and macro scales.

In a first part of this work, we extend such PGD-based time decomposition from [54,55] to the non-linear setting of cyclic elasto-plasticity. In particular, we show that such decomposition correctly performs when assuming history-dependent nonlinear behaviors, as plasticity promotes. However, here, the integration of plasticity does

not gain from the multi-time format of the time scheme and the evaluation of the nonlinear term still requires the reconstruction of the past history. This compromises the full computation in case of cyclic fatigue, thus so far the technique shares the typical shortcomings of conventional methodologies in computational plasticity.

In a second part of the work, we push the technique a step further towards long-term simulations involving a high number of cycles, focusing on the nonlinear constitutive relations. To this purpose, we combine the time-multiscale characterization with suitable data-driven techniques, establishing a predictor-corrector scheme. Predictor-corrector schemes are commonly employed also in cycle-jumping techniques [149–152]. In fact, in usual cycle-jumping methods, the extrapolated state is employed as the initial state for future finite element simulations, which are used as reference within the correction step. The drawback of such approach is that, being incremental in the predictions, the committed error is accumulated.

On the contrary, the predictor-corrector scheme here proposed is not incremental. This is achieved (a) treating exclusively the macrotime functions via the predictorcorrector scheme; (b) accounting for the spatial functions through sparse sampling and data completion techniques; (c) assuming unchanged the microtime functions and, afterwards, correcting them via a successive enrichment.

As a last introductory comment, the strategy ensures the equilibrium globally in space and time. All the stages of the procedure are based on iterative schemes whose solutions' quality is determined and, if necessary, enhanced according to suitable convergence criteria, guaranteeing robustness.

For the sake of simplicity, the chapter is divided into two sections. Section 3.2 is dedicated to the computation of a time-multiscale response in cyclic elasto-plasticity. Section 3.3 exploits the results of section 3.2 to build an efficient data-driven model of the nonlinear constitutive relations. The sections are structured as follows.

- 1. Section 3.2. Subsection 3.2.1 presents the problem statement in its strong and weak forms. Subsection 3.2.2 recasts the problem in the numerical framework of the proper generalized decomposition, starting from the space and time separation and then addressing the multi-time separation. Subsection 3.2.3 shows the results on load-unload tensile tests. Subsection 3.2.4 gives conclusions and perspectives.
- 2. Section 3.3. Subsection 3.3.1 briefly recalls the theoretical and numerical framework already introduced in section 3.2. Subsection 3.3.2 enters in the details of all the methods exploited to build the data-driven model. Subsection 3.3.3 shows the numerical results considering two benchmark tests in 2D, varying the geometry and plasticity law. Finally, subsection 3.3.4 provides conclusions and perspectives.

3.2 Multi-time PGD in cyclic elasto-plasticity

3.2.1 Theoretical framework

Neglecting the time derivatives, a quasi-static (nonlinear) problem can be expressed as

$$\mathcal{L}(u(\boldsymbol{x},t)) = f(\boldsymbol{x},t) \tag{3.2.1}$$

where $\mathcal{L}(\bullet)$ refers to a generic nonlinear differential operator involving the derivatives in space. Here the time dependence is associated to the loading $f(\boldsymbol{x}, t)$.

As shown in [54–56], when using the PGD, the solution of (3.2.1) can efficiently be computed in the separated space/multi-time form as soon as \mathcal{L} is a linear operator.

In case of nonlinearities, the nonlinear operator \mathcal{L} can be decomposed in a linear part \mathcal{L}_{l} and a nonlinear one \mathcal{L}_{nl} . In such a way, problem (3.2.1) can be rewritten as (3.2.2)

$$\mathcal{L}_{l}(u(\boldsymbol{x},t)) = f(\boldsymbol{x},t) - \mathcal{L}_{nl}(u(\boldsymbol{x},t)), \qquad (3.2.2)$$

which is easily linearized, for instance as

$$\mathcal{L}_{l}(u^{(l)}(\boldsymbol{x},t)) = f(\boldsymbol{x},t) - \mathcal{L}_{nl}(u^{(l-1)}(\boldsymbol{x},t)), \qquad (3.2.3)$$

where the superscript (l) refers to the nonlinear iteration. The solution of the linearized problem can be computed using a full space/micro-time/macro-time decomposition [54, 55]

$$u^{(l)}(\boldsymbol{x},t) \approx \sum_{k} U_{k}^{\boldsymbol{x}}(\boldsymbol{x}) U_{k}^{\tau}(\tau) U_{k}^{T}(T), \qquad (3.2.4)$$

or starting with a space/time separation and imposing a further multi-time decomposition for the computation of the time function (micro/macro time sub-modes)

$$u^{(l)}(\boldsymbol{x},t) \approx \sum_{k} U_{k}^{\boldsymbol{x}}(\boldsymbol{x}) U_{k}^{t}(t) \approx \sum_{k} U_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j} U_{k,j}^{\tau}(\tau) U_{k,j}^{T}(T).$$
(3.2.5)

In approximation (3.2.5), each time function is expressed in terms of micro-macro time sub-modes, as also suggested in [56]. However, while in [56] a symmetric-antisymmetric characterization is assumed for the sub-modes, here they are simply defined in the standard PGD manner, like in [54, 55].

The aim of this work is computing such a time multiscale representation when dealing with cyclic elasto-plasticity, where the non-linear term depends on the past history, that is

$$\mathcal{L}_{nl}(u^{(l-1)}(\boldsymbol{x},t)) = \mathcal{N}(u^{(l-1)}(\boldsymbol{x},s); s \le t),$$
(3.2.6)

where \mathcal{N} denotes a nonlinear operator.

PROBLEM STATEMENT

As a reference problem¹, the quasi-static elasto-plastic equations (under small deformations) are considered. To this purpose, let us introduce the body \mathcal{B} , occupying the spatial region $\Omega \subset \mathbb{R}^d$, with d = 2, 3, whose boundary is denoted as $\Gamma = \partial \Omega$. The body is subject to a cyclic loading $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ applied over the time interval $I = (0, T_f)$, as sketched in figure 3.1.

The variable to be determined are the displacement field $\boldsymbol{u}(\boldsymbol{x},t)$ and the stress field $\boldsymbol{\sigma}(\boldsymbol{x},t)$, with $(\boldsymbol{x},t) \in \Omega \times I$, fulfilling

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f} & \text{in } \Omega \times I \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D \times I \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{f}_N & \text{on } \Gamma_N \times I \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{in } \Omega \times \{0\} \end{cases}$$
(3.2.7)

¹Multi-dimensional functions up to second-order tensors will be denoted with bold italic letters.



FIGURE 3.1: Mechanical problem under study.

where $\mathbf{f} = \mathbf{f}(\mathbf{x}, t)$ is a prescribed history of body forces, $\Gamma_D = \partial \Omega_D$ and $\Gamma_N = \partial \Omega_N$ are the Dirichlet and Neumann regions of the boundary $\Gamma = \Gamma_D \dot{\cup} \Gamma_N$ (where the symbol $\dot{\cup}$ denotes a disjoint union) and \mathbf{n} is the outward unit normal vector to Γ_N . As usual, \mathbf{u}_0 is the initial condition, \mathbf{u}_D is a prescribed displacement on Γ_D and \mathbf{f}_N is a prescribed traction (per unit deformed area) on Γ_N .

Moreover, u and σ verify the elasto-plastic constitutive relation:

$$\boldsymbol{\sigma} = \mathsf{C} : (\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}^p(\boldsymbol{u})) \tag{3.2.8}$$

with C the fourth-order stiffness tensor, $\boldsymbol{\varepsilon} = \nabla^s \boldsymbol{u}$ the total strain tensor ($\nabla^s(\boldsymbol{\bullet})$ being the symmetric gradient operator), $\boldsymbol{\varepsilon}^p$ the plastic strain tensor and : referring to the tensor product twice contracted.

GLOBAL EQUILIBRIUM WEAK FORM

The weak formulation of the quasi-static problem (3.2.7), under small deformations hypotheses, is easily retrieved from the local form of the equilibrium [126, 153, 154].

As first, let us denote with V the space of regular enough and kinematically admissible displacement fields (i.e., ensuring the imposed displacement on the region Γ_D):

$$V = \{ \boldsymbol{u} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{u}(\boldsymbol{\gamma}, t) = \boldsymbol{u}_D(\boldsymbol{\gamma}, t), \quad (\boldsymbol{\gamma}, t) \in \Gamma_D \times I \}.$$
(3.2.9)

Similarly, let V_0 be the space of test functions, satisfying null Dirichlet condition on Γ_D , that is

$$V_0 = H^1_{0,\Gamma_D}(\Omega; \mathbb{R}^d) = \{ \boldsymbol{v} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{v}(\boldsymbol{\gamma}) = \boldsymbol{0}, \quad \boldsymbol{\gamma} \in \Gamma_D \}.$$
 (3.2.10)

With these definitions made, the weak formulation of problem (3.2.7) reads as follows: find $\boldsymbol{u}(t) \in V$ verifying

$$\int_{\Omega} [\boldsymbol{\sigma}(t) : \boldsymbol{\varepsilon}(\boldsymbol{v}) - \boldsymbol{f}(t) \cdot \boldsymbol{v}] d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f}_N(t) \cdot \boldsymbol{v} d\boldsymbol{\gamma} = 0, \qquad (3.2.11)$$

 $\forall t \in I \text{ and } \forall v \in V_0.$

By using equation (3.2.8) and reordering the terms, equation (3.2.11) becomes:

$$\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathsf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x} - \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathsf{C} : \boldsymbol{\varepsilon}^{p}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x} = \int_{\Omega} \boldsymbol{f}(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{x} + \int_{\Gamma_{N}} \boldsymbol{f}_{N}(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{\gamma},$$
(3.2.12)

or, equivalently,

$$k(\boldsymbol{u}(t), \boldsymbol{v}) - f^{p}(\boldsymbol{u}(t), \boldsymbol{v}) = f^{ext}(\boldsymbol{v}; t), \qquad (3.2.13)$$

after having introduced the bilinear and linear forms

$$\begin{cases} k(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\mathsf{C}} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x} \\ f^{ext}(\boldsymbol{v}; t) = \int_{\Omega} \boldsymbol{f}(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{f}_N(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{\gamma}, \end{cases}$$
(3.2.14)

as well as the nonlinear term accounting for the plastic strain

$$f^{p}(\boldsymbol{u},\boldsymbol{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathsf{C} : \boldsymbol{\varepsilon}^{p}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x}.$$
(3.2.15)

Let us recall that evaluation of $f^p(\boldsymbol{u}(t), \boldsymbol{v})$ in equation (3.2.13) requires the knowledge of $\boldsymbol{\varepsilon}^p(\boldsymbol{u}(t))$, which depends on the assumed plasticity model [126, 153, 154].

PLASTICITY MODEL

Here a standard von Mises (J_2) plasticity, whose yield surface is defined by

$$\Phi(\boldsymbol{\sigma}(t), \boldsymbol{\varepsilon}^p(t)) = \sqrt{3J_2} - \sigma_{y,t} = 0, \qquad (3.2.16)$$

where $J_2 = J_2(t)$ denotes the second deviatoric invariant

$$J_2 = J_2(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s} : \boldsymbol{s}, \quad \boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\sigma})\boldsymbol{I}$$
(3.2.17)

and $\sigma_{y,t}$ is the uniaxial yield stress, which evolves through a suitable strain-hardening curve. The details of this evaluation are given below.

Making the assumption of isotropic hardening, at any state of hardening, the evolution of the yield surface (3.2.16) corresponds to a uniform (isotropic) expansion of the initial one. This is obtained assuming $\sigma_{y,t}$ being a function of the accumulated (or effective) plastic strain

$$\bar{\varepsilon}_t^p = \int_0^t \sqrt{\frac{2}{3}} \dot{\varepsilon}^p : \dot{\varepsilon}^p \mathrm{d}s.$$
(3.2.18)

In particular, assuming linear hardening, $\sigma_{y,t}$ is given by

$$\sigma_{y,t} = \sigma_{y,0} + H\bar{\varepsilon}_t^p, \qquad (3.2.19)$$

where $\sigma_{y,0}$ is the initial yield stress and H is the (constant) hardening modulus.

Additionally, a standard associative plastic flow rule is considered, meaning that the plastic strain rate is a tensor normal to the yield surface in the stress space, that is

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda} \boldsymbol{N}, \quad \boldsymbol{N} := \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}.$$
 (3.2.20)

Following usual notations, in equation (3.2.20), $\dot{\lambda}$ denotes the unknown plastic multiplier. Specifically, when considering an associative hardening rule, it holds [126]

$$\dot{\varepsilon}^p = \sqrt{\frac{2}{3}} \|\dot{\varepsilon}^p\| = \dot{\lambda}. \tag{3.2.21}$$

3.2.2 MATERIAL AND METHODS

In most finite element based approaches, the equilibrium (3.2.13) is restored incrementally [126, 153]. This corresponds to a global loop, where the nonlinearity due to f^p is tackled either explicitly or implicitly. However, nonlinear problems such as (3.2.13) have also been successfully addressed using non-incremental strategies in time, as suggested by the LATIN and PGD literature [56, 155–158]. In this work, the PGD-based procedure is adopted, expressing the solution in the low-rank separated form

$$\boldsymbol{u}_m(\boldsymbol{x},t) = \sum_{k=1}^m \boldsymbol{U}_k^{\boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{U}_k^t(t), \qquad (3.2.22)$$

and computed directly in the whole space-time domain, by means of an iterative strategy. In equation (3.2.22), m denotes the rank of the solution, also known as number of PGD modes.

As usual in computational plasticity, the integration of the constitutive equations corresponds to a local loop, usually referred as state-updating procedure [126, 153]. Indeed, the plasticity model reduces to differential constitutive equations which may be solved numerically by means of an Euler scheme (explicit or implicit). Here, an implicit algorithm based on the elastic predictor/return-mapping procedure is adopted, where the resulting nonlinear equation (for the incremental plastic multiplier) is tackled by a Newton-Raphson scheme.

Let us start by describing the multi-time separation in subsection 3.2.2 before introducing the linearization strategy and the space-time separation in subsection 3.2.2.

MULTI-TIME SEPARATION

Following the same strategy of [54–56], the computation of the PGD time modes $\{U_k^t(t)\}_{k=1}^m$ in (3.2.22) can be addressed via a multi-time separated representation (MT-PGD).

In particular, the PGD solution of the linearized problem (the index of the nonlinear iteration is suppressed for notational simplicity) may be approximated via a spacemicrotime-macrotime separated representation [54,55]

$$\boldsymbol{u}_m(\boldsymbol{x},t) = \sum_{k=1}^m \boldsymbol{U}_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^t(t) \approx \sum_{k=1}^M \boldsymbol{U}_k^{\boldsymbol{x}}(\boldsymbol{x}) U_k^{\boldsymbol{\tau}}(\tau) U_k^{\boldsymbol{T}}(T), \qquad (3.2.23)$$

or imposing the multi-time decomposition for the computation of the time function (micro/macro time sub-modes) [55, 56]

$$\boldsymbol{u}_{m}(\boldsymbol{x},t) = \sum_{k=1}^{m} \boldsymbol{U}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) U_{k}^{t}(t) \approx \sum_{k=1}^{m} \boldsymbol{U}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j=1}^{m_{k}} U_{k,j}^{\tau}(\tau) U_{k,j}^{T}(T), \quad (3.2.24)$$

where two new independent time coordinates τ (microtime, or fast time) and T (macrotime, or slow time) to be introduced. Moreover, in equation (3.2.23) the new number of modes is denoted as M (in general, M > m), while in equation (3.2.24) the number of modes involved in the space-time separation is still m and m_k denotes the number of time sub-modes required to approximate the single-scale function for the current global mode k. Remark 3.1. As discussed in [54–56], if the single-scale function $U_k^t(t)$ exhibits a multiscale behaviour and a wise choice of the macro-partitions is done (e.g., respecting typical patterns of the function such as the ones coming from a periodic behaviour), only a few sub-modes m_k are enough to have a good approximation (3.2.24). Since the primary goal of this work is to investigate multiscale patterns inside PGD time functions, using representation (3.2.24) is enough.

The procedure starts by taking into account the following (macro) decomposition of the time interval I:

$$0 = T_0 < T_1 < \dots < T_{N_T - 1} < T_{N_T} = T_f \tag{3.2.25}$$

and, for each $1 \leq l \leq N_T$, assuming the following (micro) decomposition of the interval $[T_{l-1}, T_l)$:

$$T_{l-1} = \tau_0(T_l) < \tau_1(T_l) < \dots < \tau_{N_\tau - 1}(T_l) < \tau_{N_\tau}(T_l) = T_l.$$
(3.2.26)

Observing that, for each $1 \leq l \leq N_T - 1$, it holds

$$\tau_{N_{\tau}}(T_l) = T_l = \tau_0(T_{l+1}), \qquad (3.2.27)$$

one can express $t \in [0, T_f)$ as

$$t = \sum_{l=1}^{N_T} \mathbb{1}_{[T_{l-1}, T_l)}(t) = \sum_{l=1}^{N_T} \sum_{s=1}^{N_\tau} \mathbb{1}_{[\tau_{s-1}(T_l), \tau_s(T_l))}(t), \qquad (3.2.28)$$

where

$$\mathbb{1}_{[a,b)}(t) = \begin{cases} 1 & \text{if } a \le t < b, \\ 0 & \text{otherwise.} \end{cases}$$
(3.2.29)

In general, a function of space and time $\boldsymbol{u}(\boldsymbol{x},t)$ can finally be written as

$$\boldsymbol{u}(\boldsymbol{x},t) = \sum_{l=1}^{N_T} \mathbb{1}_{[T_{l-1},T_l)}(t) \boldsymbol{u}(\boldsymbol{x},t) = \sum_{l=1}^{N_T} \sum_{s=1}^{N_\tau} \mathbb{1}_{[\tau_{s-1}(T_l),\tau_s(T_l))}(t) \boldsymbol{u}(\boldsymbol{x},t), \quad (3.2.30)$$

where the two expressions $\mathbb{1}_{[T_{l-1},T_l)}(t)\boldsymbol{u}(\boldsymbol{x},t)$ and $\mathbb{1}_{[\tau_{s-1}(T_l),\tau_s(T_l))}(t)\boldsymbol{u}(\boldsymbol{x},t)$ represent \boldsymbol{u} by using slow (macro) and fast (micro) time, respectively.

Now, fixing $\boldsymbol{x} \in \Omega$ a multi-time discretization scheme is defined by

$$\mathbb{1}_{[\tau_{s-1}(T_l),\tau_s(T_l))}(t)\boldsymbol{u}(\boldsymbol{x},t) \approx \boldsymbol{u}(\boldsymbol{x},\tau_{s-1}(T_l)) \in \mathbb{R}^k, \qquad (3.2.31)$$

for $1 \leq l \leq N_T$, $1 \leq s \leq N_\tau$ and k = 2, 3.

As a final remark, the slow and fast decompositions of the time axis furnish a matrix representation $\mathbf{t} = (t_{s,l})_{s=1,l=1}^{s=N_{\tau},l=N_T} \in \mathbb{R}^{N_{\tau} \times N_T}$ of the time t, through

$$t_{s,l} := \tau_{s-1}(T_l). \tag{3.2.32}$$

LINEARIZATION AND SPACE-TIME SEPARATION

Following a standard Galerkin approach, the approximation of u(t) satisfying (3.2.13) is sought in the finite dimensional subspace of V defined as

$$V^{h} = \left\{ \boldsymbol{u}_{h} \in V \text{ s.t. } \boldsymbol{u}_{h}(\boldsymbol{x},t) = \sum_{i=1}^{N_{\boldsymbol{x}}} \boldsymbol{u}_{i}(t)\phi_{i}(\boldsymbol{x}), \quad (\boldsymbol{x},t) \in \Omega \times I \right\}$$
(3.2.33)

where $\{\phi_i\}_{i=1}^{N_x}$ is the set of suitably chosen shape functions in space and $\{u_i(t)\}_{i=1}^{N_x}$ the corresponding unknown temporal coefficients. Similarly, let us denote with V_0^h the Galerkin approximation space of V_0 .

Within the semi-discrete counterpart of (3.2.13), one looks for $\boldsymbol{u}_h \in V^h$ such that, $\forall t \in I \text{ and } \forall \boldsymbol{v}_h \in V_0^h$,

$$k(\boldsymbol{u}_h(t), \boldsymbol{v}_h) - f^p(\boldsymbol{u}_h(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t).$$
(3.2.34)

To tackle the nonlinearity of the problem, the first step consists in computing an approximation of the elastic solution verifying², $\forall t \in I$

$$k(\boldsymbol{u}^{(0)}(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t), \qquad (3.2.35)$$

whose algebraic counterpart is straightforwardly obtained as

$$\mathbf{Ku}^{(0)}(t) = \mathbf{f}^{ext}(t), \qquad (3.2.36)$$

where $K_{ij} = k(\phi_i, \phi_j)$ and $f_i^{ext}(t) = f^{ext}(\phi_i; t)$, for $i, j = 1, \dots, N_x$.

At this point, a temporal discretization of the interval I is introduced. This is based on considering N_t uniform times t_i , such that $t_{i+1} - t_i = \Delta t > 0$, for $i = 1, \ldots, N_t$.

Equation (3.2.36) can then be rewritten in a tensorial formalism over the whole spacetime domain [54], looking for $\mathbf{U}^{(0)} \in \mathbb{R}^{N_x \times N_t}$ such that

$$(\mathbf{K} \otimes \mathbf{I}_{N_t}) : \mathbf{U}^{(0)} = \mathbf{F}^{ext}$$
(3.2.37)

where $\mathbf{I}_{N_t} \in \mathbb{R}^{N_t \times N_t}$ denotes the identity matrix in time and $\mathbf{F}^{ext} \in \mathbb{R}^{N_x \times N_t}$ collects the time evaluations of the right-hand-side in (3.2.36), that is $\mathbf{f}^{ext}(t_i) \in \mathbb{R}^{N_x}$, $i = 1, \ldots, N_t$. Problem (3.2.37) can be treated directly by means of the PGD algorithm, which seeks a low-rank separated approximation of $\mathbf{U}^{(0)}$ as

$$\mathbf{U}^{(0)} \approx \mathbf{U}_{m^{(0)}}^{(0)} = \sum_{k=1}^{m^{(0)}} \mathbf{U}_{k}^{(0), \boldsymbol{x}} \otimes \mathbf{U}_{k}^{(0), t}, \qquad (3.2.38)$$

where $m^{(0)}$ is the number of PGD modes. For the PGD assembly and solution of space-time separated problem, the reader may refer to [54] and references therein.

Since $\mathbf{U}_{m^{(0)}}^{(0)}$ is a low-rank approximation of $\boldsymbol{u}^{(0)}(t)$, $\forall t \in I$. An approximation of the elastic solution $\boldsymbol{u}^{(0)}(t)$, $\forall t \in I$ all history-dependent variables may be updated from $\boldsymbol{\varepsilon}^{(0)} = \nabla^s \boldsymbol{u}^{(0)}$. In particular, the update of $\boldsymbol{\varepsilon}^{p,(0)}$ allows to freeze the nonlinear term in equation (3.2.13) and start an iterative process where each iteration $l \geq 1$ consists of two steps:

1. (*state-updating*) The update of $\varepsilon^{p,(l-1)} = \varepsilon^p(\boldsymbol{u}^{(l-1)})$ via the elastic predictor/return-mapping procedure. This consists of a set of nonlinear evaluations which can be written as

$$\boldsymbol{\varepsilon}^{p,(l-1)} = \mathcal{N}(\boldsymbol{\varepsilon}^{(l-1)}, \bar{\boldsymbol{\varepsilon}}^p_{T_{\boldsymbol{\varepsilon}}}), \qquad (3.2.39)$$

where \mathcal{N} represents the nonlinear operator depending on the total strain tensor and on the effective plastic strain up to the final time T_f .

²For the sake of notational simplicity, the subscript h related to the Galerkin approximation is suppressed.

2. (linearized equilibrium) The solution of the linearized problem

$$k(\boldsymbol{u}^{(l)}(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t) + f^p(\boldsymbol{u}^{(l-1)}(t), \boldsymbol{v}_h), \qquad (3.2.40)$$

looking for the space-time separated approximation

$$\mathbf{U}^{(l)} \approx \mathbf{U}_{m^{(l)}}^{(l)} = \sum_{k=1}^{m^{(l)}} \mathbf{U}_{k}^{(l), x} \otimes \mathbf{U}_{k}^{(l), t}$$
(3.2.41)

of the corresponding tensorial problem

$$(\mathbf{K} \otimes \mathbf{I}_{N_t}) : \mathbf{U}^{(l)} = \mathbf{F}^{ext} + \mathbf{F}^{p,(l-1)}$$
(3.2.42)

with $\mathbf{F}^{p,(l-1)} \in \mathbb{R}^{N_x \times N_t}$ accounting for the plastic contributions.

The linearization loop stops when two successive approximations become close enough under a suitable distance. This means that, given a small enough $\delta > 0$

$$e_{l} = \frac{\left\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\right\|_{F}}{\left\|\mathbf{U}^{(l-1)}\right\|_{F}} < \delta,$$
(3.2.43)

where $\|\bullet\|_F$ denotes, for instance, the standard Frobenius norm.

Denoting with L the iteration satisfying the convergence criterion (3.2.43), the final PGD approximation of the nonlinear problem (3.2.13) is $\mathbf{U}^{(L)}$, that is the approximation (3.2.22) has rank $m = m^{(L)}$.

The overall solving procedure is summarized in the flowchart in figure 3.2.

FIGURE 3.2: PGD-based solving scheme for elasto-plasticity.

Remark 3.2. The assembly of the stiffness matrix **K** is a classic finite element based one and the evaluation of right-hand-side $\mathbf{F}^{p,(l-1)}$ follows a standard plasticity integration algorithm (elastic predictor/return-mapping procedure). Both these computations can be performed using any computational mechanics software. The overall procedure is thus weakly-intrusive, since it only requires the externalization of the linearization loop and the usage of the space-time PGD solver after the assembly.

When accounting for the multi-time decomposition introduced in subsection 3.2.2, as widely discussed in [54], the usual single-scale time grid $\{t_i\}_{i=0}^{N_t}$, with $N_t = N_T N_\tau$ becomes a matrix of dimension $\mathbb{R}^{N_T \times N_\tau}$ recovered by the tensor product of two newly introduced micro and macro grids. Letting $\{\tau_s^0\}_{s=0}^{N_\tau}$ the micro-grid defined along the first macro interval $[T_0, T_1)$, it suffices to multiply it tensorially by the macro-dofs $\{T_l\}_{l=0}^{N_T-1}$ to recover all the degrees of freedom in time, that is $\{T_l\}_{l=0}^{N_T-1} \otimes \{\tau_s^0\}_{s=0}^{N_\tau}$.

Within the PGD assembly (3.2.37), when considering time multiscale approximations, the time operator is exactly recovered as $\mathbf{I}_{N_t} = \mathbf{I}_{N_T} \otimes \mathbf{I}_{N_{\tau}}$ (when considering PDEs involving time derivatives, exact tensorial decompositions of the related time operators have been discussed in [54]).

In this multiscale formalism, a discrete function of time $\mathbf{h}^t \in \mathbb{R}^{N_T \times N_\tau}$ is now expressed as

$$\mathbf{h}^t \approx \sum_{j=1}^{m_k} \mathbf{h}_j^T \otimes \mathbf{h}_j^\tau, \qquad (3.2.44)$$

and, particularly, the PGD approximation (3.2.41) is replaced by its MT-PGD counterpart

$$\mathbf{U}^{(l)} \approx \mathbf{U}_{m^{(l)}}^{(l)} = \sum_{k=1}^{m^{(l)}} \mathbf{U}_{i}^{(l), \boldsymbol{x}} \otimes \sum_{j=1}^{m_{k}} \mathbf{U}_{k, j}^{(l), T} \otimes \mathbf{U}_{k, j}^{(l), \tau}.$$
 (3.2.45)

This straightforwardly entails a reduced time multiscale representation of the total strain

$$\boldsymbol{\varepsilon}_{m}(\boldsymbol{x},t) = \sum_{k=1}^{m} \boldsymbol{\varepsilon}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \boldsymbol{\varepsilon}_{k}^{t}(t) \approx \sum_{k=1}^{m} \boldsymbol{\varepsilon}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j=1}^{m_{k}} \boldsymbol{\varepsilon}_{k,j}^{\tau}(\tau) \boldsymbol{\varepsilon}_{k,j}^{T}(T).$$
(3.2.46)

Remark 3.3. In approximation (3.2.45), each product $\mathbf{U}_{k,j}^{(l),T} \otimes \mathbf{U}_{k,j}^{(l),\tau}$ is not granted to be continuous by construction. The tensor product is, indeed, replicating the microscale patterns $\mathbf{U}_{k,j}^{(l),\tau}$ along the macroscale. However, the continuity of the multitime approximation of $\mathbf{U}^{(l)}$ is ensured by adding a sufficient number of modes m_k .

Remark 3.4. As observed in [54], the usage of multi-time representations guarantees reduced storage requirements since N_t function evaluations are reconstructed combining N_T and N_{τ} evaluations of the macro and micro functions. Moreover, as usual in PGD methods, the size of independent systems to be solved in the alternating direction strategy (ADS) is smaller when employing the time separation, implying computational savings.

Remark 3.5. It shall be noticed that, so far, the computational gains entailed by the usage of the MT-PGD are the ones discussed in [54,56] for linear problems. Indeed, the nonlinearity the step 1 (state-updating) requires a standard integration over N_t steps because of the history-dependency. In other terms, so far, the MT-PGD is not exploited for the evaluation of the elastoplastic constitutive relation. In the work-flow in figure 3.2, the PGD solver blocks are replaced by the MT-PGD ones. The computational effectiveness is being addressed in some works in progress, where the

time-multiscale representation is exploited for an efficient evaluation of the nonlinear term.

3.2.3 **Results and Discussion**

In this section, two examples in cyclic elasto-plasticity are considered, varying the geometry of the specimen \mathcal{B} and the imposed loading u_D .

The problems are solved numerically via the PGD space-time constructor and special care is paid to the PGD time modes and their multi-time characterization (MT-PGD).

The equations correspond to a slight simplification of (3.2.7):

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = 0 & \text{in } \Omega \times I \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \Gamma_D \times I \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{0} & \text{on } \Gamma_N \times I \\ \boldsymbol{u} = \boldsymbol{0} & \text{in } \Omega \times \{0\} \end{cases}$$
(3.2.47)

with $\Omega \subset \mathbb{R}^2$ and $u_D = (u_D(t), 0)$, that is a uniaxial load-unload tensile test. The boundary Γ_D corresponds to the left and right sides of the specimen, while Γ_N to the upper and lower sides.

Moreover, 2D plane strain is considered, meaning that the Hooke's law (3.2.8) simplifies to

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ 2\varepsilon_{12} \end{pmatrix}.$$
(3.2.48)

All the numerical values of model and material parameters are specified in the subsections below.

Dog-bone shaped specimen

A uniaxial load-unload tensile test over a 2D dog-bone shaped steel specimen under monoperiodic cyclic loading is here considered. The loading consists in a Dirichlet datum $u_D(t)$ having constant amplitude applied to both sides of the specimen.

The material has a Young's modulus E = 210 GPa and a Poisson's ratio $\nu = 0.3$. The assumed plasticity law (3.2.19) is characterized by an initial yield stress $\sigma_{y,0} = 205$ MPa and a linear hardening coefficient H = 2 GPa. To ensure small deformations, the imposed constant amplitude displacement has a maximum amplitude $u_D^{max} = 0.125$ mm and the load rate is fixed at $v_l = 0.025$ mm/s following the standard of quasi-static testing. A single cycle (load-unload-load) time has duration $T_1 = 4u_D^{max}/v_l = 20$ s.

Figure 3.3 shows the two-dimensional discretized geometry and the imposed displacement having 10 cycles. The spatial mesh consists of $N_e = 500$ quadrilateral elements and $N_x = 561$ mesh nodes. The time interval is divided in $N_t = 800$ times.

Figure 3.4 gives the magnitude of the displacement field and the isotropic hardening function computed at the final time $T_f = 200$ s.

Figure 3.5 shows the comparison of the PGD results with a classical FE-based incremental algorithm (considering implicit integration of plasticity based on returnmapping) computed at the center of the specimen $(x_0, y_0) = (0, 0)$.



FIGURE 3.3: Discretized geometry (left) and imposed displacement (right).



FIGURE 3.4: Displacement field (left) and isotropic hardening (right) at final time T_f .



FIGURE 3.5: Stress-displacement curve (left) and hysteresis loop (right) in (0,0). Red line: FE, black dashed line: PGD.



FIGURE 3.6: First four normalized PGD modes.

The first four normalized modes in space and time of the PGD approximation (3.2.22) are shown in figure 3.6.

Focusing on the time modes, one can observe that U_1^t corresponds to the elastic response, while subsequent modes exhibit similar evolution, characterized by (a) a transient zone at the beginning, (b) a pattern stabilization towards an almost-periodic (cyclic) behavior, (c) a slow decay of the signals amplitude. This behaviour can be easily characterized by the MT-PGD approximation (3.2.24):

$$\boldsymbol{u}_{m}(\boldsymbol{x},t) = \sum_{k=1}^{m} \boldsymbol{U}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) U_{k}^{t}(t) \approx \sum_{k=1}^{m} \boldsymbol{U}_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j=1}^{m_{k}} U_{k,j}^{\tau}(\tau) U_{k,j}^{T}(T).$$
(3.2.49)

This is achieved through a macro-discretization based on N_T equispaced macro-times, while the micro-scale consists of N_{τ} equispaced micro-times. For instance, if a single cycle is defined from the sequence loading-unloading-loading, the macro-discretization could consists (as later commented out, without losing generality) of a coarse mesh having a macro timestep ΔT covering a whole cycle. The micro-discretization corresponds to a fine mesh along the cycle, as illustrated in figure 3.7.



FIGURE 3.7: Microscale and macroscale time discretization.

The advantage of considering such a decomposition is evident when increasing the number of cycles. For instance, figure 3.8 is the counterpart of the time modes of figure 3.6 when imposing the same loading over 60 cycles.



FIGURE 3.8: PGD time modes $\{U_k^t(t)\}_{k=1}^4$ with 60 cycles.

When employing the MT-PGD, each one of the signals in figure 3.8 is approximated in terms of micro-macro submodes. For instance, the top of figure 3.9 shows the second mode $U_2^t(t)$ computed via the PGD algorithm and by its multi-time counterpart MT-PGD (black dashed and blue lines superposed as shown in the zoomed figure). The images at bottom show the micro-time modes $\{U_{2,j}^\tau(\tau)\}_{j=1}^4$ and macro-time ones $\{U_{2,j}^T(T)\}_{j=1}^4$, respectively.



FIGURE 3.9: First four micro-macro modes of the multi-time decomposition of U_2^t .

As shown in figure 3.9, the micro-macro characterization is strongly physically consistent with the previously highlighted evolution of the time response. Indeed, the microscale functions capture the almost-periodicity of the function, through cyclic highly nonlinear patterns exhibiting fast dynamics. On the contrary, the dynamics is really slow along the macroscale, whose functions present an initial transitorial behavior, followed by a smooth evolution.

Let us briefly comment about the choice of macropartitions (i.e., the choice of ΔT) in the definition of the macroscale. As formerly discussed in [54], the tensorial decomposition beyond the multi-time strategy makes this choice completely arbitrary, however results may be affected. As observed in [54], if the choice is not physically meaningful, the convergence of the multiscale approximation may be exacerbated (more time submodes might be required to achieve convergence).

In order to achieve optimal convergence, a first concern is thus establishing a physically consistent decomposition of the time domain. For instance, referring to figure 3.7, a wise solution could be choosing $\Delta T = kt_C$, with $k \in \mathbb{N}^+$, meaning that ΔT is a multiple of the external excitation period. Afterwards, the choice of k may be driven by both physical and computational reasons. On the one hand, it may depend on what one aims at capturing along the fast scale (for instance, k = 1 corresponds to a full cycle response). On the other hand, the separated problems arising within the submodes computation are scaling with N_{τ} and N_T for the micro and macro scales, respectively. Letting N be the number of cycles, one gets $N_T = N/k$ and $N_{\tau} = N_t/N_T$, therefore the choice of k affects their complexities.

PLATE WITH EDGE CRACK

A second application case concerns a 2D plate of the same material, having an edge crack in the upper part, represented by the red segment in the left-side of figure 3.10.

Moreover, we consider a displacement imposed over 40 cycles with a given slope (no more centered in zero), as shows the right-side of figure 3.10. The spatial mesh consists of $N_e = 400$ quadrilateral elements and $N_x = 451$ nodes. The time interval is divided in $N_t = 3200$ times. A single cycle (load-unload-load) time has duration $T_1 = 30$ s, meaning that the final time is $T_f = 1200$ s.



FIGURE 3.10: Discretized geometry (left) and imposed displacement (right).

Figure 3.11 gives the magnitude of the displacement field and the isotropic hardening function computed at the final time $T_f = 1200$ s.



FIGURE 3.11: Displacement field (left) and isotropic hardening (right) at final time T_f .

Figure 3.12 shows the comparison of the PGD results with a classical FE-based incremental algorithm (considering implicit integration of plasticity based on returnmapping) computed at the center of the specimen $(x_0, y_0) = (0, 0)$.



FIGURE 3.12: Stress-displacement curve (left) and hysteresis loop (right) in (0,0). Red line: FE, black dashed line: PGD.

The first four normalized modes in space and time of the PGD approximation (3.2.22) are shown in figure 3.13.



FIGURE 3.13: First four normalized PGD modes.

The time modes in figure 3.13 clearly exhibit a multiscale behavior, easily recognized when employing the MT-PGD. As for the previous example, the number of macrodofs corresponds to the number of cycles $N_T = 40$, while the microscale has $N_{\tau} = 80$ times spanning the first macro-interval (in high-cycle analyses, a larger ΔT could be preferred to reduce the complexity of the macroscale problem). The multi-time decomposition of the second PGD time mode U_2^t is illustrated in figure 3.14.



FIGURE 3.14: First four micro-macro modes of the multi-time decomposition of U_2^t .

As observed from figure 3.9, also in figure 3.14 one can observe that the repeating patterns are identified by the microscale modes, while their stabilization is slower and retrieved through the macroscale modes, which track the almost linear trend at large scale.

3.2.4 CONCLUSIONS

As an extension of [54], this study successfully accomplished the computation of timeseparated solutions in the nonlinear setting of cyclic elasto-plasticity.

Within the PGD-based time multiscale procedure, the time response is computed along two separated time scales, the micro and the macro one. Such scales are defined as newly independent coordinates, while the full scale is recovered via their tensor product. The study shows physically consistent results for the elasto-plastic response under cyclic loading. Indeed, delegating a whole cycle to the microscale, highly nonlinear patterns are observed over the fast scale, while a smooth and slow evolution is captured by the macroscale. This makes the macroscale characterization particularly attractive for long-time horizon analyses, such as aging and fatigue [159].

In [56], authors propose an efficient algorithm to optimally decompose complex signals (involving many frequencies) in a fast and slow scale. Prior to the multi-time approximation, the algorithm from [56] may be applied to the external excitation of the problem at hand. In such a way, an optimal decomposition of the time axis may be established apriori (i.e., the best value of the macroscale step size ΔT is determined), guaranteeing the optimal convergence of the multi-time PGD procedure. Works in progress are dealing with this topic.

For what concerns the nonlinear character of the problem, the linearization procedure is based on solving over the full space/time domain the elastic problem and enforcing the plastic contribution to the right-hand-side. The integration of plasticity is performed through the elastic predictor/return-mapping algorithm [126, 153]. This can be viewed as a nonlinear operator \mathcal{N} acting on the total strain tensor and on the effective plastic strain up to the final time T_f , that is

$$\boldsymbol{\varepsilon}^{p,(l-1)} = \mathcal{N}(\boldsymbol{\varepsilon}^{(l-1)}, \bar{\boldsymbol{\varepsilon}}^p_{T_f}). \tag{3.2.50}$$

Specifically, the evaluation of (3.2.50) requires the reconstruction over the whole past history since

$$\bar{\varepsilon}_t^p = \int_0^t \sqrt{\frac{2}{3}} \dot{\varepsilon}^p : \dot{\varepsilon}^p \mathrm{d}s.$$
(3.2.51)

In this sense, so far, the proposed methodology shares the usual drawbacks of standard techniques in computational plasticity since the integration of plasticity does not benefit from the multi-time format of the time scheme.

In section 3.3 we focus on this matter. In particular, we exploit the multiscale representation to build an efficient data-driven model of the elasto-plastic constitutive relation. This would allow real-time evaluations of (3.2.50), enabling the direct simulation of inelasticity also in long-term scenarios, such as high-cycle fatigue.

As a final comment, the method benefits of the usual advantages entailed by PGDbased procedures. For instance, model parameters and loading conditions can be treated as problem extra-coordinates, enabling the fast computation of multiparametric solutions [160–162]. Moreover, the further time separation guarantees, when solving the linearized problem, the same operational and memory savings discussed in [54].

3.3 MULTI-TIME BASED LEARNING OF HISTORY-DEPENDENT NONLINEAR BEHAVIORS

3.3.1 THEORETICAL AND NUMERICAL FRAMEWORK

By denoting with $\mathcal{L}(\bullet)$ a generic nonlinear differential operator involving the space derivatives, the addressed quasi-static problem can be written as

$$\mathcal{L}(u(\boldsymbol{x},t)) = f(\boldsymbol{x},t), \qquad (3.3.1)$$

where the time dependence is associated to the cyclic loading $f(\boldsymbol{x}, t)$. The nonlinear operator \mathcal{L} is decomposed additively into a linear and a nonlinear part, as $\mathcal{L} = \mathcal{L}_{l} + \mathcal{L}_{nl}$. If the superscript (l) tracks the nonlinear iteration, problem (3.3.1) can be linearized as

$$\mathcal{L}_{l}(u^{(l)}(\boldsymbol{x},t)) = f(\boldsymbol{x},t) - \mathcal{L}_{nl}(u^{(l-1)}(\boldsymbol{x},t)), \qquad (3.3.2)$$

whose solution may be computed in the multi-time form [56]

$$u^{(l)}(\boldsymbol{x},t) \approx u^{(l)}(\boldsymbol{x},\tau,T) = \sum_{k} U_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \sum_{j} U_{k,j}^{\tau}(\tau) U_{k,j}^{T}(T).$$
(3.3.3)

where τ denotes the time microscale variable and T the macroscale one (this has been widely discussed in section 3.2).

This task is computationally cheap using the standard PGD constructor [30, 155, 163-165], even when considering parameters [161, 162, 166, 167], and becomes even faster when making use of multi-time separated representations in equation (3.3.2), that is

$$f(\boldsymbol{x},t) \approx \sum_{j} F_{j}^{\boldsymbol{x}}(\boldsymbol{x}) F_{j}^{\tau}(\tau) F_{j}^{T}(T), \quad \mathcal{L}_{\mathrm{nl}}(u^{(l-1)}(\boldsymbol{x},t)) \approx \sum_{j} L_{j}^{\boldsymbol{x}}(\boldsymbol{x}) L_{j}^{\tau}(\tau) L_{j}^{T}(T).$$
(3.3.4)

Such expressions may be obtained, among other possibilities, via the higher-order SVD (HOSVD) [168, 169] or the PGD [163].

However, as pointed out in section 3.2, the calculation of $\mathcal{L}_{nl}(u^{(l-1)}(\boldsymbol{x},t))$ becomes a tricky issue when

$$\mathcal{L}_{nl}(u^{(l-1)}(\boldsymbol{x},t)) = \mathcal{N}(u^{(l-1)}(\boldsymbol{x},s); s \le t)$$
(3.3.5)

where \mathcal{N} denotes a nonlinear operator. According to (3.3.5), the nonlinearity is local in space but history-dependent in time, as encountered in elasto-plastic behaviors in solid mechanics. For instance, keeping the same notation of section 3.2 where hardening plasticity is considered, a nonlinear operator \mathcal{N} acts on the total strain tensor and on the effective plastic strain up to the final time T_f , that is

$$\boldsymbol{\varepsilon}^{p,(l-1)} = \mathcal{N}(\boldsymbol{\varepsilon}^{(l-1)}, \bar{\boldsymbol{\varepsilon}}^p_{T_f}). \tag{3.3.6}$$

Specifically, the evaluation of (3.3.6) requires the reconstruction over the whole past history since

$$\bar{\varepsilon}_t^p = \int_0^t \sqrt{\frac{2}{3}} \dot{\varepsilon}^p : \dot{\varepsilon}^p \,\mathrm{d}s. \tag{3.3.7}$$

Thus, the construction of the right-hand side entails two main difficulties common to all standard discretization techniques: (i) because of the behavior locality, the nonlinear term must be evaluated at each location \boldsymbol{x} used for discretizing equation (3.3.2); (ii) the nonlinear term must be evaluated along the whole (long) time interval with the resolution enforced by the fastest physics, that is τ . These requirements of course compromise the solution of problems defined over large time intervals $I = (0, T_f)$.

Here below we shortly recall the theoretical and numerical aspects of the analyzed problem, already introduced in 3.2.

PROBLEM STATEMENT

The reference problem is the same treated in section 3.2, consisting of an elasto-plastic structure occupying the spatial region Ω and subject to a cyclic loading $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{x},t)$ applied over the time interval $I = (0, T_f)$. The unknowns are the displacement field $\boldsymbol{u}(\boldsymbol{x},t)$ and the stress field $\boldsymbol{\sigma}(\boldsymbol{x},t)$, with $(\boldsymbol{x},t) \in \Omega \times I$, satisfying

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f} & \text{in } \Omega \times \boldsymbol{I} \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \partial \Omega_D \times \boldsymbol{I} \\ \boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{f}_N & \text{on } \partial \Omega_N \times \boldsymbol{I} \\ \boldsymbol{u} = \boldsymbol{u}_0 & \text{in } \Omega \times \{0\}. \end{cases}$$
(3.3.8)

Using standard notations, \boldsymbol{u}_0 is the initial condition, \boldsymbol{u}_D is a prescribed displacement on $\partial \Omega_D$ and \boldsymbol{f}_N is a prescribed traction (per unit deformed area) on $\partial \Omega_N$.

Moreover, u and σ verify the elasto-plastic constitutive relation

$$\boldsymbol{\sigma} = \mathsf{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \tag{3.3.9}$$

with C the fourth-order stiffness tensor, $\boldsymbol{\varepsilon} = \nabla^s \boldsymbol{u}$ the total strain tensor ($\nabla^s(\boldsymbol{\bullet})$ being the symmetric gradient operator), $\boldsymbol{\varepsilon}^p$ the plastic strain tensor and : referring to the tensor product twice contracted.

PLASTICITY MODEL

The yield surface is defined by a standard von Mises (J_2) plasticity as

$$\Phi(\boldsymbol{\sigma}(t), \boldsymbol{\varepsilon}^p(t)) = \sqrt{3J_2} - \sigma_{y,t} = 0, \qquad (3.3.10)$$

where $J_2 = J_2(t)$ denotes the second deviatoric invariant

$$J_2 = J_2(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s} : \boldsymbol{s}, \quad \boldsymbol{s} = \boldsymbol{\sigma} - \frac{1}{3}\operatorname{tr}(\boldsymbol{\sigma})\boldsymbol{I}$$
(3.3.11)

and $\sigma_{y,t}$ is the uniaxial yield stress, which evolves through a suitable strain-hardening curve. The details of this evaluation are given below.

Making the assumption of isotropic hardening, at any state of hardening entails $\sigma_{y,t}$ being a function of the effective plastic strain

$$\bar{\varepsilon}_t^p = \int_0^t \sqrt{\frac{2}{3}} \dot{\varepsilon}^p : \dot{\varepsilon}^p \,\mathrm{d}s. \tag{3.3.12}$$

In particular, among many other possibilities [126,153], a Voce-type nonlinear isotropic hardening can be assumed

$$\sigma_{y,t} = \sigma_{y,0} + H\bar{\varepsilon}_t^p + (\sigma_\infty - \sigma_0)(1 - \exp^{-\delta\bar{\varepsilon}_t^p}), \qquad (3.3.13)$$

where $\sigma_{y,0}$ is the reference yield stress, H is the Voce's linear hardening modulus, σ_{∞} is the limit stress parameter, and δ is the Voce hardening parameter.

Additionally, a standard associative plastic flow rule is considered, meaning that the plastic strain rate is a tensor normal to the yield surface in the stress space, that is

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda} \boldsymbol{N}, \quad \boldsymbol{N} := \frac{\partial \Phi}{\partial \boldsymbol{\sigma}} = \sqrt{\frac{3}{2}} \frac{\boldsymbol{s}}{\|\boldsymbol{s}\|}.$$
 (3.3.14)

Following usual notations, in equation (3.3.14), λ denotes the unknown plastic multiplier. Specifically, when considering an associative hardening rule, it holds [126]

$$\dot{\bar{\varepsilon}}^p = \sqrt{\frac{2}{3}} \|\dot{\varepsilon}^p\| = \dot{\lambda}. \tag{3.3.15}$$

WEAK FORMULATION

Before switching to the numerical framework, problem (3.3.8) is recast in its weak form. To this purpose, let V be the space of regular enough and kinematically admissible displacement fields

$$V = \{ \boldsymbol{u} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{u}(\boldsymbol{\gamma}, t) = \boldsymbol{u}_D(\boldsymbol{\gamma}, t), \quad (\boldsymbol{\gamma}, t) \in \Gamma_D \times I \}$$
(3.3.16)

and let V_0 be the space of test functions, satisfying null Dirichlet condition on Γ_D ,

$$V_0 = H^1_{0,\Gamma_D}(\Omega; \mathbb{R}^d) = \{ \boldsymbol{v} \in H^1(\Omega; \mathbb{R}^d) : \boldsymbol{v}(\boldsymbol{\gamma}) = \boldsymbol{0}, \quad \boldsymbol{\gamma} \in \Gamma_D \}.$$
(3.3.17)

One can define (see section 3.2) the following bilinear and linear forms

$$\begin{cases} k(\boldsymbol{u}, \boldsymbol{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\mathsf{C}} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x} \\ f^{ext}(\boldsymbol{v}; t) = \int_{\Omega} \boldsymbol{f}(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{x} + \int_{\partial \Omega_N} \boldsymbol{f}_N(t) \cdot \boldsymbol{v} \mathrm{d}\boldsymbol{\gamma}, \end{cases}$$
(3.3.18)

and the nonlinear term accounting for the plastic strain

$$f^{p}(\boldsymbol{u},\boldsymbol{v}) = \int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathsf{C} : \boldsymbol{\varepsilon}^{p}(\boldsymbol{u}) \mathrm{d}\boldsymbol{x}.$$
(3.3.19)

At this point, the weak formulation consists of seeking $\boldsymbol{u}(t) \in V$ verifying

$$k(\boldsymbol{u}(t), \boldsymbol{v}) - f^p(\boldsymbol{u}(t), \boldsymbol{v}) = f^{ext}(\boldsymbol{v}; t), \qquad (3.3.20)$$

 $\forall t \in I \text{ and } \forall v \in V_0.$

NUMERICAL APPROXIMATION

The numerical approximation of $\boldsymbol{u}(t)$ satisfying (3.3.20) is sought in the Galerkin approximation space defined as

$$V^{h} = \left\{ \boldsymbol{u}_{h} \in V \text{ s.t. } \boldsymbol{u}_{h}(\boldsymbol{x},t) = \sum_{i=1}^{N_{\boldsymbol{x}}} \boldsymbol{u}_{i}(t)\phi_{i}(\boldsymbol{x}), \quad (\boldsymbol{x},t) \in \Omega \times I \right\}$$
(3.3.21)

where $\{\phi_i\}_{i=1}^{N_x}$ is the set of suitably chosen shape functions in space and $\{u_i(t)\}_{i=1}^{N_x}$ the corresponding unknown temporal coefficients. Similarly, let us denote with V_0^h the approximation space of V_0 .

Equation (3.3.20) is rewritten in its semi-discrete counterpart, where one looks for $\boldsymbol{u}_h \in V^h$ such that, $\forall t \in I$ and $\forall \boldsymbol{v}_h \in V_0^h$,

$$k(\boldsymbol{u}_h(t), \boldsymbol{v}_h) - f^p(\boldsymbol{u}_h(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t).$$
(3.3.22)

The nonlinear problem (3.3.22) is solved in a non-incremental manner exploiting the PGD. The procedure is based on a global space-time iterative solver, where the first step consists of neglecting the plastic contribution (nonlinear term) and computing the so-called elastic mode. This means finding $\boldsymbol{u}^{(0)}(t)$ such that, $\forall t \in I$,

$$k(\boldsymbol{u}^{(0)}(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t),$$
 (3.3.23)

whose algebraic counterpart reads

$$\mathbf{Ku}^{(0)}(t) = \mathbf{f}^{ext}(t), \qquad (3.3.24)$$

where $K_{ij} = k(\phi_i, \phi_j)$ and $f_i^{ext}(t) = f^{ext}(\phi_i; t)$, for $i, j = 1, \dots, N_x$.

Now, the temporal interval I is then discretized in N_t uniform times t_i , such that $t_{i+1} - t_i = \Delta t > 0$, for $i = 1, ..., N_t$, and equation (3.3.24) recast in a tensorial formalism over the whole space-time domain [54]. In this context, one seeks $\mathbf{U}^{(0)} \in \mathbb{R}^{N_x \times N_t}$ such that

$$(\mathbf{K} \otimes \mathbf{I}_{N_t}) : \mathbf{U}^{(0)} = \mathbf{F}^{ext} \tag{3.3.25}$$

where $\mathbf{I}_{N_t} \in \mathbb{R}^{N_t \times N_t}$ denotes the identity matrix in time and $\mathbf{F}^{ext} \in \mathbb{R}^{N_x \times N_t}$ collects the time evaluations of the right-hand-side in (3.3.24), that is $\mathbf{f}^{ext}(t_i) \in \mathbb{R}^{N_x}$, $i = 1, \ldots, N_t$. The solution $\mathbf{U}^{(0)}$ of problem (3.3.25) can be computed using the PGD method [54], which expresses $\mathbf{U}^{(0)}$ in the separated form

$$\mathbf{U}^{(0)} \approx \mathbf{U}_{m^{(0)}}^{(0)} = \sum_{k=1}^{m^{(0)}} \mathbf{U}_{k}^{(0), \boldsymbol{x}} \otimes \mathbf{U}_{k}^{(0), t}, \qquad (3.3.26)$$

where $m^{(0)}$ is the rank of the approximation (number of modes).

Once the approximation of $\boldsymbol{u}^{(0)}(t)$, $\forall t \in I$, is known, all history-dependent variables are updated from $\boldsymbol{\varepsilon}^{(0)} = \nabla^s \boldsymbol{u}^{(0)}$ and a linearization scheme is set-up. This consists of the following two steps:

1. (*state-updating*) The integration of plasticity via an implicit algorithm (elastic predictor/return-mapping procedure) [126, 153], which can be summarized as

$$\boldsymbol{\varepsilon}^{p,(l-1)} = \mathcal{N}(\boldsymbol{\varepsilon}^{(l-1)}, \bar{\boldsymbol{\varepsilon}}^p_{T_{\boldsymbol{\varepsilon}}}), \qquad (3.3.27)$$

where \mathcal{N} is a nonlinear operator depending on the total strain tensor and on the effective plastic strain up to the final time T_f .

2. (linearized equilibrium) The solution of the linearized problem

$$k(\boldsymbol{u}^{(l)}(t), \boldsymbol{v}_h) = f^{ext}(\boldsymbol{v}_h; t) + f^p(\boldsymbol{u}^{(l-1)}(t), \boldsymbol{v}_h), \qquad (3.3.28)$$

which is expressed in a tensorial form

$$(\mathbf{K} \otimes \mathbf{I}_{N_t}) : \mathbf{U}^{(l)} = \mathbf{F}^{ext} + \mathbf{F}^{p,(l-1)}$$
(3.3.29)

and whose solution is computed by means of the usual PGD solver as

$$\mathbf{U}^{(l)} \approx \mathbf{U}_{m^{(l)}}^{(l)} = \sum_{k=1}^{m^{(l)}} \mathbf{U}_{k}^{(l), \boldsymbol{x}} \otimes \mathbf{U}_{k}^{(l), t}.$$
 (3.3.30)

Notice that, with respect to (3.3.25), in problem (3.3.29), the updated nonlinear contribution $\mathbf{F}^{p,(l-1)} \in \mathbb{R}^{N_x \times N_t}$ appears to the right-hand-side.

The procedure stops when the successive enrichment error is small enough with respect to a given tolerance $\delta > 0$, that is

$$e_{l} = \frac{\left\|\mathbf{U}^{(l)} - \mathbf{U}^{(l-1)}\right\|_{F}}{\left\|\mathbf{U}^{(l-1)}\right\|_{F}} < \delta,$$
(3.3.31)

where $\|\bullet\|_F$ denotes, for instance, the standard Frobenius norm.

Denoting with L the iteration satisfying the convergence criterion (3.3.31), the final PGD approximation of the nonlinear problem (3.3.20) is $\mathbf{U}^{(L)}$, having rank $m = m^{(L)}$. The overall solving procedure is recalled in figure 3.15.



FIGURE 3.15: PGD solving scheme for elasto-plasticity.

The present section focuses on the red box in figure 3.15, which means the evaluation of the nonlinear constitutive relations. Indeed, when addressing a high-number of cycles, this step becomes unfeasible, due to memory and computational issues. Here, rhe novel contribution stands in proposing a data-driven modeling of the nonlinear relations with the aim of accelerating such step. Moreover, its originality comes from the usage of a time multiscale characterization to build efficiently such model, as detailed in subsection 3.3.2.

3.3.2 Multiscale-based data-driven modeling

Let us suppose that the nonlinear evaluation (3.3.27) can be performed up to $K \ll N$ cycles. This allows to compute the nonlinear terms $\varepsilon^{p,(l-1)}$ over the space-time domain $\Omega \times I_K$, with $I_K = (0, T_K]$, where T_K denotes the endpoint of the K-th loading cycle. Denoting with T_N the endpoint of the N-th loading cycle, the aim of the data-driven modeling is to forecast the nonlinear term over $\hat{I} = (T_K, T_N]$ without additional computational costs³.

Exploiting the multi-time PGD constructor [54, 56], the suggested strategy starts by decomposing the space-time evolution of $\varepsilon^{p,(l-1)}$ in slow and fast time dynamics, via the separated approximation

$$\boldsymbol{\varepsilon}^{p,(l-1)} \approx \sum_{k=1}^{m} \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_k^{\tau}(\tau) \Psi_k^{T}(T).$$
(3.3.32)

In approximation (3.3.32), a generic function of the microscale $\Psi_k^{\tau}(\tau)$ exhibits a complex highly nonlinear behaviour due to the plasticity occuring over the short scale. On the contrary, a function $\Psi_k^T(T)$ of the macroscale is characterized by a really smooth evolution, enabling the accurate and efficient prediction of the long-term evolution. The macrotime predictions are then inserted into a predictor-corrector workflow, as illustrated by the scheme⁴ in figure 3.16.



FIGURE 3.16: Workflow of the time-multiscale based data-driven approach.

The workflow consists of five main blocks: (a) performing the nonlinear evaluations up to T_K and computing its multi-time approximation; (b) forecasting the macrotime evolution; (c) predicting the nonlinear response up to T_N using the macrotime forecast; (d) correcting the prediction integrating the nonlinear relations in a few spatial locations; (e) considering the predicted-corrected nonlinear evolution to assemble the linearized problem up to T_N .

As a matter of fact, the time-multiscale decomposition provides a characterization of the history-dependent nonlinear behavior, greatly simplifying the machine learningbased forecasting task, which would otherwise be very difficult and expensive to com-

³Similarly, the hat $\hat{\bullet}$ notation will be reserved for the predicted quantities over \hat{I} .

⁴In the scheme the superscript (l-1) has been dropped for notational simplicity.

pute if left to a full data-based time integrator. In this sense, the procedure can be interpreted in the framework of physics-based machine learning approaches [170].

The following sections explain in detail the predictor-corrector scheme, corresponding to the steps (c) and (d).

PREDICTOR

The predictor is built separately for each macrotime mode $v = \Psi_k^T(T)$, k = 1, ..., m(scalar-valued function), whose corresponding snapshot (time series) can be written as $\mathbf{v} = (v_1, ..., v_{N_T}) \in \mathbb{R}^{N_T}$. The number of data points coincides with the number of macrodofs N_T and the sampling interval is the macro time step ΔT .

Exploiting only the macro functions has several computational advantages. A few of them are listed here below.

- 1. The size of the analyzed snapshots is reduced. If N_{τ} is the number of dofs along the microscale and N_T the number of dofs along the macro one, the length of the time signals reduces from $N_t = N_T N_{\tau}$ encountered in classical time marching schemes to N_T .
- 2. The smooth behavior along the macroscale entails further compression of the snapshots, guaranteeing more memory savings. Indeed,
 - (a) the macro modes may be well characterized by means of a few shape parameters p allowing highly-accurate reconstructions (e.g., low-order polynomials) of the signal over all the steps N_T ;
 - (b) a resampling of the macro modes based on $N'_T \ll N_T$ steps will not loose accuracy in the approximation, since all the high frequencies are tracked by the micro modes.
- 3. Forecasting along the macroscale is a much easier task for any time integrator, since all the patterns and highly nonlinear evolution are delegated to the microscale modes.

Among many other possibilities [85,171–174], this work adopts the higher-order DMD for the time series forecasting. The dynamic mode decomposition (DMD) [175] is a well known snapshots-based technique allowing to extract relevant patterns in nonlinear dynamics, closely related to the Koopman theory [176–178]. The higher-order DMD (HODMD) is an extension of the former, which considers time-lagged snapshots [179, 180]. This technique is particularly attractive for the purposes of this work due to its ability of allowing rich extrapolations involving nonzero decaying rates [179].

The algorithm beyond the HODMD is also called DMD-*d* algorithm, since it considers *d*-lagged elements. For $d \ge 1$ fixed hyper-parameter, this means that the following higher-order Koopman assumption is made [179]

$$v_{j+d} \approx c_1 v_j + c_2 v_{j+1} + \dots + c_d v_{j+d-1}, \tag{3.3.33}$$

which is rewritten in terms of standard Koopman assumption as

$$\tilde{\mathbf{v}}_{j+1} \approx \tilde{\mathbf{R}} \tilde{\mathbf{v}}_j, \tag{3.3.34}$$

involving enlarged snapshots and (unknown) Koopman matrix

$$\tilde{\mathbf{v}}_{j} = \begin{pmatrix} v_{j} \\ v_{j+1} \\ \vdots \\ v_{j+d-2} \\ v_{j+d-1} \end{pmatrix} \in \mathbb{R}^{d}, \quad \tilde{\mathbf{R}} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ c_{1} & c_{2} & c_{3} & \cdots & c_{d-1} & c_{d} \end{pmatrix} \in \mathbb{R}^{d \times d}, \quad (3.3.35)$$

with $1 \le j \le N_T - d$.

All the implementation details of the HODMD are given in section A.3.

Once the HODMD-based models for the macroscale modes $\{\Psi_k^T(T)\}_{k=1}^m$ are trained, they give the predictions $\{\hat{\Psi}_k^T(T)\}_{k=1}^m$ over \hat{I} . Re-using the microscale and spatial modes from (3.3.32), the nonlinear response is predicted over \hat{I} as

$$\hat{\boldsymbol{\varepsilon}}^{p,l-1}(\boldsymbol{x},t)\mathbb{1}_{\hat{I}}(t) = \sum_{k=1}^{m} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x})\Psi_{k}^{\tau}(\tau)\hat{\Psi}_{k}^{T}(T).$$
(3.3.36)

This is schematically illustrated in figure 3.17, where ψ denotes the nonlinear response $\varepsilon^{p,l-1}$ particularized in a spatial location.



FIGURE 3.17: Macrotime forecast and response prediction through the microtime patters.

Remark 3.3.1. As discussed in section 3.2, the response $\psi(\boldsymbol{x},t) = \boldsymbol{\varepsilon}^{p,(l-1)}$ may exhibit an initial transient behaviour (i.e., micro-scale patterns evolving in time). Practically, this acerbates the micro-macro separated representation (3.3.32), requiring more modes to describe the transient zone. To avoid these additional modes, prior the multitime approximation of $\psi(\boldsymbol{x},t)$, one should determine a time instant $T_K > T_i > 0$ from which the response can be assumed stabilized in terms of microscale patterns.

Such time T_i is efficiently determined, for instance, using the gappy-POD [181–183], whose continuous approximation of $\psi(\boldsymbol{x}, t)$ can be written as (all details are given in appendix A.2.1)

$$\psi(\boldsymbol{x},t) \approx \psi^{\text{GPOD}}(\boldsymbol{x},t) = \sum_{k=1}^{m} \bar{w}_k(\boldsymbol{x})\bar{\alpha}_k(t),$$
 (3.3.37)

with \bar{m} denoting the number of gappy-POD modes, while $\bar{w}_k(\boldsymbol{x})$ and $\bar{\alpha}_k(t)$ are the space and time functions, respectively [56]. The time modes $\bar{\alpha}_k(t)$ allow to quickly identify an approximation of the threshold T_i .

As soon as T_i is computed, approximation (3.3.32) is replaced by (3.3.38)

$$\psi(\boldsymbol{x}, t) \mathbb{1}_{(T_i, T_K)}(t) \approx \sum_{k=1}^m \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_k^{\boldsymbol{\tau}}(\tau) \Psi_k^{\boldsymbol{T}}(T), \qquad (3.3.38)$$

where $\mathbb{1}_{I}(t)$ denotes the indicator function of the set I.

Afterwards, a data-driven model on the macroscale modes is trained, following the procedure detailed in section 3.3.2, where the higher-order DMD is considered. The stabilized response and its multiscale decomposition is, for instance, illustrated in figure 3.18.



FIGURE 3.18: Transient zone and stabilized response.

CORRECTOR

The quality of the prediction (3.3.36) should be compared with a full integration of the plasticity up to T_N , that is

$$\boldsymbol{\varepsilon}^p = \mathcal{N}(\boldsymbol{\varepsilon}, \bar{\varepsilon}^p_{T_N}). \tag{3.3.39}$$

However, as already discussed, evaluations in (3.3.39) are unfeasible when $T_N \gg T_K$ and when too many spatial nodes N_x are considered. Let us assume that this task, however, can be performed locally for a few reference spatial locations $\mathbf{x}_r = \{\mathbf{x}_1^r, \ldots, \mathbf{x}_J^r\}$, with $1 < J < N_x$, like in sparse-sampling-based approaches (the locations can, for instance, be selected as the ones having the highest accumulated plastic strain $\bar{\varepsilon}_{T_K}^p$).

In this sparse framework, instead of considering (3.3.39), the correction of $\hat{\varepsilon}^{p,(l-1)}$ is based on employing its reduced counterpart over the set \mathbf{x}_r , which can be denoted as

$$\boldsymbol{\varepsilon}_{\mathbf{x}_r}^p = \mathcal{N}_{\mathbf{x}_r}(\boldsymbol{\varepsilon}, \bar{\varepsilon}_{T_N}^p). \tag{3.3.40}$$

The predictor (3.3.36) is corrected updating the macro modes by solving the following minimization problem:

$$\min_{\{\Delta \Psi_k^T\}_{k=1}^m} \left\| \sum_{k=1}^m \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_k^{\tau}(\tau) \Big(\hat{\Psi}_k^T(T) + \Delta \Psi_k^T(T) \Big) - \boldsymbol{\varepsilon}^p(\boldsymbol{x}, t) \right\|_{\Omega_r \times \hat{I}}, \quad (3.3.41)$$

where $\|\bullet\|_{\Omega_r \times \hat{I}} = \int_{\hat{I}} \int_{\Omega_r} \bullet d\mathbf{x} dt$ denotes a norm suitably defined over the reduced spatial domain Ω_r and the temporal prediction interval \hat{I} .

Problem (3.3.41) can be recast in a weighted residual form, after having introduced suitable test functions $\{\Phi_k^T\}_{k=1}^m$, by

$$\int_{\Omega_r \times \hat{I}} \left(\sum_{k=1}^m \Psi_k^x(\boldsymbol{x}) \Psi_k^\tau(\tau) \Phi_k^T(T) \right) \left(\sum_{l=1}^m \Psi_l^x(\boldsymbol{x}) \Psi_l^\tau(\tau) \Delta \Psi_l^T(T) - \hat{e}(\boldsymbol{x}, t) \right) \mathrm{d}\boldsymbol{x} \mathrm{d}t = 0, \quad (3.3.42)$$

where $\hat{e}(\boldsymbol{x},t)$ simply corresponds to the prediction error function, which can be expressed into a time-separated form after having rearranged $\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t)$ as $\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},\tau,T)$:

$$\hat{e}(\boldsymbol{x},\tau,T) = \sum_{k=1}^{m} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_{k}^{\tau}(\tau) \hat{\Psi}_{k}^{T}(T) - \psi(\boldsymbol{x},\tau,T).$$
(3.3.43)

The following integrals can be defined, for all k, l = 1, ..., m,

$$a_{kl} = \left(\int_{\Omega_r} \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_l^{\boldsymbol{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}\right) \left(\int_{I^\tau} \Psi_k^{\tau}(\tau) \Psi_l^{\tau}(\tau) \mathrm{d}\tau\right)$$
(3.3.44)

and

$$b_k(T) = \int_{I^\tau} \left(\int_{\Omega_r} \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \hat{e}(\boldsymbol{x},\tau,T) \mathrm{d}\boldsymbol{x} \right) \Psi_k^{\tau}(\tau) \mathrm{d}\tau.$$
(3.3.45)

With these definitions made, equation (3.3.42) can be rewritten as

$$\int_{\hat{I}^T} \sum_{k=1}^m \Phi_k^T(T) \sum_{l=1}^m \Delta \Psi_l^T(T) a_{kl} \mathrm{d}T = \int_{\hat{I}^T} \sum_{k=1}^m \Phi_k^T(T) b_k(T) \mathrm{d}T.$$
(3.3.46)

At this point, problem (3.3.46) can be easily solved using finite elements in time, among other possibilities.

In the above definitions, the time intervals I^{τ} and \hat{I}^{T} are the ones associated to the micro and macro scales, respectively. In particular, the one related to the macroscale keeps the hat notation since it concerns the forecasting interval.

ENRICHMENT

Once the optimal macrotime correction modes $\{\Delta \Psi_k^T(T)\}_{k=1}^m$ satisfying (3.3.46) have been determined, a global enrichment step can be performed. This consists in adding ulterior modes $m^\star - m - 1$ to enrich the PGD approximation, solving the minimization problem

$$\min_{\{\Psi_k^x, \Psi_k^\tau, \Psi_k^T\}_{k=m+1}^{m^*}} \left\| \sum_{k=m+1}^{m^*} \Psi_k^x(\boldsymbol{x}) \Psi_k^\tau(\tau) \Psi_k^T(T) - \hat{e}^{\text{update}}(\boldsymbol{x}, \tau, T) \right\|_{\Omega_r \times \hat{I}}, \quad (3.3.47)$$

where

$$\hat{e}^{\text{update}}(\boldsymbol{x},\tau,T) = \boldsymbol{\varepsilon}^{p}(\boldsymbol{x},\tau,T) - \sum_{k=1}^{m} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_{k}^{\tau}(\tau) \Big(\hat{\Psi}_{k}^{T}(T) + \Delta \Psi_{k}^{T}(T) \Big).$$
(3.3.48)

The minimization problem can be rewritten in the following weighted residual form

$$\int_{\Omega_r \times \hat{I}} \sum_{k=m+1}^{m^*} \Phi_k(\boldsymbol{x}, \tau, T) \left(\sum_{k=m+1}^{m^*} \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_k^{\tau}(\tau) \Psi_k^{T}(T) - \hat{e}^{\text{update}}(\boldsymbol{x}, \tau, T) \right) = 0. \quad (3.3.49)$$

In problem (3.3.49) the following test function has been introduced

$$\Phi_k(\boldsymbol{x},\tau,T) = \Phi_k^{\boldsymbol{x}}(\boldsymbol{x})\Psi_k^{\tau}(\tau)\Psi_k^{T}(T) + \Psi_k^{\boldsymbol{x}}(\boldsymbol{x})\Phi_k^{\tau}(\tau)\Psi_k^{T}(T) + \Psi_k^{\boldsymbol{x}}(\boldsymbol{x})\Psi_k^{\tau}(\tau)\Phi_k^{T}(T), \quad (3.3.50)$$

where Φ_k^x, Φ_k^τ and Φ_k^T are three independent test functions, for the space, micro time and macrotime problems, respectively. Finally, the solution of (3.3.49) is obtained by means of a fixed-point alternating direction strategy, as usual in PGD-based procedures [155, 163].

The corrected (optimal) predictor, after the update-enrichment procedure is then defined as

$$\hat{\varepsilon}^{p,\star}(\boldsymbol{x},t) = \underbrace{\sum_{k=1}^{m} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_{k}^{\tau}(\tau) \left(\hat{\Psi}_{k}^{T}(T) + \Delta \Psi_{k}^{T}(T)\right)}_{\text{update}} + \underbrace{\sum_{k=m+1}^{m^{\star}} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_{k}^{\tau}(\tau) \Psi_{k}^{T}(T)}_{\text{enrichment}}.$$
 (3.3.51)

Remark 3.3.2. Let us add a remark concerning the computation of the spatial functions $\Psi_k^x(\boldsymbol{x})$ over the whole domain Ω . Indeed, the minimization problem (3.3.47) is defined only over the reduced domain Ω_r involving the reference spatial locations $\mathbf{x}_r = \{\boldsymbol{x}_1^r, \ldots, \boldsymbol{x}_J^r\}$, with $1 < J < N_x$. Therefore, the gappy-POD spatial bases of the approximation (3.3.37) are used to reconstruct the functions on Ω . Let us recall the gappy-POD approximation (3.3.37)

$$\psi(\boldsymbol{x},t) \approx \psi^{\text{GPOD}}(\boldsymbol{x},t) = \sum_{k=1}^{\bar{m}} \bar{w}_k(\boldsymbol{x})\bar{\lambda}_k(t).$$
 (3.3.52)

The spatial basis is collected in the matrix

$$\bar{\mathbf{W}} = \begin{pmatrix} \bar{\mathbf{w}}_1 | & \cdots & | \bar{\mathbf{w}}_{\bar{m}} \end{pmatrix} \in \mathbb{R}^{N_x \times \bar{m}}, \qquad (3.3.53)$$

where the column vectors $\bar{\mathbf{w}}_k$ are the discretized counterpart of the spatial modes $\bar{w}_k(\mathbf{x})$.

The reduced counterpart of (3.3.53) over Ω_r can be denoted as $\bar{\mathbf{W}}_r \in \mathbb{R}^{J \times \bar{m}}$ and is obtained by selecting the lines corresponding to the locations in \mathbf{x}_r .

Given a PGD spatial mode $\Psi_k^x(\boldsymbol{x})$, for $k = 1, \ldots, m$, known over the reduced domain Ω_r , it can be written in vector form as $\boldsymbol{\psi}_k^x \in \mathbb{R}^{J \times 1}$. Its reconstruction on Ω via gappy-POD is simply obtained as

$$\boldsymbol{\psi}_{k}^{\boldsymbol{x},\text{GPOD}} = \bar{\mathbf{W}}\boldsymbol{\alpha}_{k}, \qquad (3.3.54)$$

where the vector of coefficients α_k is obtained solving the usual minimization problem

$$\min_{\boldsymbol{\alpha}_k \in \mathbb{R}^{\bar{m} \times 1}} \left\| \bar{\mathbf{W}}_r \boldsymbol{\alpha}_k - \boldsymbol{\psi}_k^{\boldsymbol{x}} \right\|_2, \tag{3.3.55}$$

with $\|\bullet\|_2$ denoting the standard Euclidean norm.

SUMMARY OF THE SOLVING SCHEME

The overall solving procedure is summarized in the flowchart in figure 3.19.

When $t \in (0, T_K]$, the algorithm consists of computing the quasi-static elasto-plastic response, using the PGD-based approach proposed in section 3.2.

When, $t \in (T_K, T_N]$, a snapshot of the plastic strain tensor $\varepsilon^p \mathbb{1}_{(0,T_K]}(t)$ is exploited to build a data-driven forecasting model of the nonlinear constitutive relations. The prediction $\hat{\varepsilon}^p \mathbb{1}_{(T_K, T_N]}(t)$ is, then, corrected by means of a sparse selection of reference spatial locations \mathbf{x}_r . Once the corrected (optimal) prediction $\hat{\varepsilon}^{p,\star} \mathbb{1}_{(T_K, T_N]}(t)$ is available, the linearized problem is assembled up to T_N and, finally, efficiently solved via the MT-PGD.



FIGURE 3.19: Data-driven MT-PGD solving scheme for elasto-plasticity.

Here below a brief recap of the computational and memory savings is given. For the sake of simplicity, the discussion considers the macroscale tracking all the cycles and the microscale evolving within a single cycle.

An incremental finite element based simulation considering N_t^C increments for single cycle and subject to N cycles (thus $N_t^C N$ increments) would require an asymptotic complexity scaling as $\mathcal{O}(N_x N_t^C N)$, where N_x is the number of spatial mesh points. The proposed procedure requires, a complexity of $\mathcal{O}(N_x N_t^C K)$, with $K \ll N$ the number of training cycles, followed by a constant negligible complexity of the HODMD- based predictor. Afterwards, the simulation extension to N cycles requires (a) the correction of the predictor based on the full-history integration over the reduced set of locations $\mathbf{x}_r = \{\mathbf{x}_1^r, \ldots, \mathbf{x}_J^r\}$, with $J \ll N_{\mathbf{x}}$, having complexity $\mathcal{O}(JN_t^CN)$, (b) the solution of the linearized problem employing the time multiscale PGD, with a complexity of $\mathcal{O}(N + N_t^C + N_{\mathbf{x}})$. This implies interesting computational gains observing the ratio $\frac{\mathcal{O}(JN_t^CN) + \mathcal{O}(N + N_t^C + N_{\mathbf{x}})}{\mathcal{O}(N_{\mathbf{x}}N_t^CN)}$, with $J \ll N_{\mathbf{x}}$.

It is worth noticing the advantages in terms of storage requirements in the final linearized problem. A usual time marching scheme requires the storage of time functions discretized in $N_t^C N$ points (where N_t^C can be really high when a small step is required). Contrarily, when employing the multi-time PGD, the functions are stored as N_t^C and N points, for the microscale and macroscale, respectively. Moreover, thanks to their slow evolution, the macro functions can be reconstructed only by means of a few coefficients p. In this case, in terms of storage one gets the ratio $\frac{p+N_t^C}{N_t^C N} \approx \frac{1}{N}$, which basically scales with the macrotime scale dimension, as already discussed in [54].

3.3.3 **Results and Discussion**

This section presents the numerical results over two different examples, chosen according to common benchmark tests in cyclic fatigue testing, which are mostly limited to simple two-dimensional specimen considering uniaxial loading [184–187]. For the validation of the PGD-based nonlinear solver in three-dimensional cases, we refer to [156], which shows the effectiveness of the method in the context of nonlinear elasto-viscoplastic behaviours encountered in polycrystalline aggregates under cyclic loading.

Dog-bone shaped specimen

For the sake of consistency with section 3.2, the first numerical example consists also of a uniaxial load-unload tensile test over a dog-bone shaped steel specimen. The loading in a Dirichlet datum $u_D(t)$ having constant amplitude applied to both sides of the specimen.

The imposed displacement has a maximum amplitude $u_D^{max} = 0.125$ mm and a single cycle (load-unload-load) time has duration $T_1 = 4u_D^{max}/v_l = 20$ s, where $v_l = 0.025$ mm/s is the load rate ensuring a quasi-statics simulation.

Young's modulus	E = 210 GPa
Poisson's coefficient	$\nu = 0.3$
Linear isotropic hardening parameters	
Reference yield stress	$\sigma_{y,0} = 205 \text{ MPa}$
Voce's linear hardening modulus	H = 2 GPa

The material parameters are set according to table 3.1.

TABLE 3.1: Material parameters for steel dog-bone shaped specimen.

Figure 3.20 shows the two-dimensional discretized geometry, consisting of $N_e = 500$ quadrilateral elements and $N_x = 561$ mesh nodes. The cyclic loading is also shown in figure 3.20. The simulation is performed using the algorithms from section 3.2 up to K = 500 cycles. The simulation is then extended in almost real-time to N = 1500 cycles using the data-driven modeling of the nonlinear term. Let us denote with T_K

and T_N the ending times of the cycles K and N, respectively. The time intervals $I_K = (0, T_K]$ and $I_N = (T_K, T_N]$ are both discretized in equispaced $N_t^{(K)} = N_t^{(N)} = 4 \cdot 10^5$ time instants.



FIGURE 3.20: Discretized geometry (left) and imposed displacement (right).

Figure 3.21 gives the magnitude of the displacement field and the isotropic hardening function computed at the time $T_K = 10^4$ s.



FIGURE 3.21: Displacement field (left) and isotropic hardening (right) at $T_f = T_K$.

The plastic strain tensor history $\varepsilon^{p} \mathbb{1}_{I_{K}}(t)$ is here used to build the data-driven model as described in Section 3.3.2. The related snapshot is defined as

$$\boldsymbol{\Psi} = \begin{pmatrix} \Psi_1 | & \cdots & | \Psi_{N_t^{(K)}} \end{pmatrix} \in \mathbb{R}^{3N_x \times N_t^{(K)}}$$
(3.3.56)

where ψ_j is a column vector containing the numerical approximations of $\boldsymbol{\varepsilon}^p(\boldsymbol{x}, t_j)$ in all the $N_{\boldsymbol{x}}$ spatial mesh points, for $j = 1, \ldots, N_t^{(K)}$. The column vectors account for the concatenation of the three components $\boldsymbol{\varepsilon}^p = (\varepsilon_{11}^p, \varepsilon_{12}^p, \varepsilon_{22}^p)$ for the two-dimensional case here analyzed.

Figure 3.21 gives the magnitude of the plastic strain computed at final time T_K .



FIGURE 3.22: Components of the plastic strain tensor $\boldsymbol{\varepsilon}^p = (\varepsilon_{11}^p, \varepsilon_{12}^p, \varepsilon_{22}^p)$ at $T_f = T_K$.

To quickly illustrate the time evolution of $\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t)$, a POD-based reduced representation [7,27] of the snapshot (3.3.56) can be considered, being the approximation

$$\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t) \approx \boldsymbol{\varepsilon}^{p,\text{POD}}(\boldsymbol{x},t) = \sum_{k=1}^{m} w_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \alpha_{k}^{t}(t), \qquad (3.3.57)$$

where the functions $w_k^{\boldsymbol{x}}(\boldsymbol{x})$ and $\alpha_k^t(t), k = 1, \dots, m$ are the space and time modes.

For instance, figure 3.23 depicts the first four POD time functions over the first 50 cycles. Even though the functions exhibit a decay towards 0, the highly nonlinear patterns (at the cycle level) make difficult the construction of a prediction model able to track accurately the fast scale. The forecasting task is clearly simplified when



FIGURE 3.23: First four normalized POD time modes of $\boldsymbol{\varepsilon}^{p,\mathrm{POD}}(\boldsymbol{x},t)$.

considering the multi-time PGD approximation (3.3.32)

$$\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t) \approx \boldsymbol{\varepsilon}^{p,\text{MT-PGD}}(\boldsymbol{x},\tau,T) = \sum_{k=1}^{m} \Psi_{k}^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_{k}^{\tau}(\tau) \Psi_{k}^{T}(T).$$
(3.3.58)

In fact, figure 3.24 shows the micro time and macro time functions, $\Psi_k^{\tau}(\tau)$ and $\Psi_k^{T}(T)$, respectively. The spatial modes $\Psi_k^{\boldsymbol{x}}(\boldsymbol{x})$ are shown in figure 3.25.

Even if the total number of retained modes is m = 6, only the first three modes were shown for the sake of conciseness.



FIGURE 3.24: First three normalized MT-PGD time modes of $\boldsymbol{\varepsilon}^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.



FIGURE 3.25: First three normalized MT-PGD space modes of $\varepsilon^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.



FIGURE 3.26: HODMD-based prediction of the macrotime modes of $\varepsilon^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.

The HODMD-based extensions of the macrotime functions are shown in figure 3.26 and used to predict the nonlinear response via (3.3.36) (the lag factor in the HODMD training was fixed to d = 40).

Letting $\|\bullet\|_{\Omega \times \hat{t}} \int_{\hat{L}} \int_{\Omega} \bullet dx dt$, the prediction error can be measured as

$$\hat{\epsilon} = \frac{\|\hat{\varepsilon}^{p}(\boldsymbol{x},t) - \boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t)\|_{\Omega \times \hat{I}}}{\|\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t)\|_{\Omega \times \hat{I}}},$$
(3.3.59)

and amounts to $\hat{\epsilon} = 0.2146$. This discrepancy is recovered through the correction step. For a better understanding of the forecasting results, let us consider the time response $\psi(t) = \varepsilon_{11}^p(\boldsymbol{x}_c, t)$ in the center of the specimen $\boldsymbol{x}_c = (0, 0)$. In figure (3.27), the reference response, the predicted signal $\hat{\psi}$ and the predicted-corrected one $\hat{\psi}^*$ are plotted in blue, red and green colors, respectively. Particularly, the loss of amplitude and the slightly inaccurate patterns of the predictor $\hat{\psi}$ are recovered by its corrected counterpart $\hat{\psi}^*$.

In this case, 8 elements (more could be selected if needed) are enough to reduce the error to $\hat{\epsilon}^* = 0.01$. The elements have been selected as those having maximum effective plastic strain $\bar{\varepsilon}_{T_K}^p$, avoiding redundant information dues to symmetry. The selected elements are plotted in red over the mesh in the upper left corner of figure 3.27, while the measurement point is the blue location \boldsymbol{x}_c . The same procedure applies when changing the imposed displacement. For instance, one can consider a linearly increasing average, as depicted in figure 3.28. Here, he red line represents the average, whose slope is u_D^{max}/T_f [m/s]. Figure 3.29 shows the HODMD-based extension of the macrotime modes and the reconstructed signal at the center of the specimen $\boldsymbol{x}_c = (0, 0)$, whose evolving patterns are accurately captured. Also in this case the corrector step utilizes the same elements as before.



FIGURE 3.27: Reference solution –blue line– compared with the predicted response $\hat{\psi}(t)$ –red line– and with the predicted-corrected one $\hat{\psi}^{\star}(t)$ –green line–, in the spatial location $\boldsymbol{x}_{c} = (0, 0)$.



FIGURE 3.28: Cyclic displacement with linearly increasing average.



FIGURE 3.29: HODMD-based modes prediction of the macrotime modes (left) and reference solution –blue line– compared with the predicted-corrected response $\hat{\psi}^{\star}(t)$ – green line–, in the spatial location $\boldsymbol{x}_{c} = (0,0)$ (right).

In terms of computational time gains, the performed tests show that the data-driven based evaluation of the nonlinear term (red versus green box in figure 3.19) has a speed-up factor of 2.3, approximately. Moreover, the overall solver time comparisons (Figure 3.15 versus figure 3.19) shows a speed-up of 2.9, approximately. The additional gain around 0.6 comes from the separated space-microtime-macrotime format of the predicted right-hand-side. Indeed, as discussed in the introduction, the PGD solver assembly becomes faster when all the terms in the equation have separated representations.

PLATE WITH HOLE

The second numerical example consists of the plate shown in figure 3.30 (left), having a circular hole of radius 1.5 cm. The mesh consists of $N_e = 400$ quadrilateral elements and $N_x = 440$ mesh nodes.

The imposed displacement, temporal discretization and choice of I_K and I_N are the same as in the previous example.



FIGURE 3.30: Discretized geometry (left) and imposed displacement (right).

The material parameters are set according to table 3.2.

Young's modulus	E = 210 GPa
Poisson's coefficient	$\nu = 0.3$
Nonlinear isotropic hardening Voce's parameters	
Reference yield stress	$\sigma_{y,0} = 152 \text{ MPa}$
Limit stress parameter	$\sigma_{\infty} = 550$ MPa
Voce's linear hardening modulus	$H=0.05~\mathrm{GPa}$
Voce hardening parameter	$\delta = 3.5$

TABLE 3.2: Material parameters for steel plate with hole.

Figure 3.31 gives the magnitude of the displacement field and the isotropic hardening function computed at the time $T_K = 10^4$ s.

Figure 3.32 gives the magnitude of the plastic strain computed at final time T_K .

For the sake of comparison with the previous example, to quickly illustrate the time evolution of $\boldsymbol{\varepsilon}^{p}(\boldsymbol{x},t)$, figure 3.33 reports the POD time functions extracted over the first 50 cycles. One can observe that, with respect to figure 3.23, here the evolution has a slower decay towards 0, which is ascribed to the nonlinear hardening law parameters.



FIGURE 3.31: Displacement field (left) and isotropic hardening (right) at $T_f = T_K$.



FIGURE 3.32: Components of the plastic strain tensor $\varepsilon^p = (\varepsilon_{11}^p, \varepsilon_{12}^p, \varepsilon_{22}^p)$ at $T_f = T_K$.



FIGURE 3.33: First four normalized POD time modes of $\boldsymbol{\varepsilon}^{p,\text{POD}}(\boldsymbol{x},t)$.

When performing the MT-PGD decomposition of the plastic strain tensor history $\varepsilon^{p} \mathbb{1}_{I_{K}}(t)$ a higher number of modes m = 8 is retained. The first three modes are plotted in figures 3.34 and 3.35.

The HODMD-based extensions of the macrotime functions give consistent results also in this case, as shown in figure 3.36 (the lag factor in the HODMD training was fixed to d = 40).

The time response $\psi(t) = \varepsilon_{11}^p(\boldsymbol{x}_c, t)$ is compared in the location $\boldsymbol{x}_c = (0, 0.015)$ (internal hole). As shown in figure 3.37, also in this case the wrong patterns and inaccurate amplitude of the predictor $\hat{\psi}$ are recovered by its corrected counterpart $\hat{\psi}^*$. The selected correction elements are plotted in red over the mesh in the upper left corner of figure 3.37, while the measurement point is the blue location \boldsymbol{x}_c . In this case, 10 elements are used to reduce the error to $\hat{\epsilon}^* = 0.01$. Here, the elements have been selected sparsely over the domain since the points exhibiting maximum effective strain in this case would coincide with the points of major physical interest around the internal circle.






FIGURE 3.35: First three normalized MT-PGD space modes of $\varepsilon^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.



FIGURE 3.36: HODMD-based prediction of the macrotime modes of $\boldsymbol{\varepsilon}^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.

DISCUSSION ABOUT HYPER-PARAMETERS

The method includes some hyper-parameters, whose choice may be problemdependent and may require specific optimization studies.

As common to all data-driven approaches, a first choice concerns the splitting of the time domain in training and predicting intervals $I_K = (0, T_K]$ and $I_N = (T_K, T_N]$. Indeed, this is mostly driven by two factors: (a) the multi-time decomposition shall exist and involve a rather small number of modes m, and (b) the extracted macro-time dynamics $\{\Psi_k^T(T)\}_{k=1}^m$ shall be learnable by the chosen integrator (HODMD-based in



FIGURE 3.37: Reference solution –blue line– compared with the predicted response $\hat{\psi}(t)$ –red line– and with the predicted-corrected one $\hat{\psi}^{\star}(t)$ –green line–, in the spatial location $\boldsymbol{x}_{c} = (0, 0.015)$.

this case).

For what concerns point (a), the micro-macro separated representation can be acerbated by an initial transient behaviour of the system response. In fact, this causes micro-scale patterns evolving in time and a high number of modes may be required to describe this zone. One option to overcome this issue is defining the training interval with respect to a lower bound $0 < T_i < T_K$ from which the patterns can be considered stabilized, at a given user-defined tolerance. A pattern stabilization detection algorithm can be defined, for instance, analyzing the POD time modes of the snapshot.

Point (b) may depend on the macrotime evolution and on the HODMD algorithm specifics. For instance, its accuracy may depend on the tuning of the lag factor d [179, 180].

Another hyper-parameter is the location and number of elements $\mathbf{x}_r = \{\mathbf{x}_1^r, \dots, \mathbf{x}_J^r\}$ chosen in the correction step. This choice may be driven by similar approaches developed within the context of hyper-reduction and sample-based methodologies [183, 188, 189].

3.3.4 CONCLUSIONS

This work aims at reducing the computational complexity of numerical simulations in cyclic loading analyses, in particular when history-dependent nonlinear behaviors are considered. To this purpose, a novel time multiscale based data-driven modeling of the nonlinearity is proposed. The procedure makes use of multi-time PGD-based representations to separate the fast (micro) and slow (macro) time dynamics.

The first step consists in collecting the plastic strain history (and other nonlinear variables evolution, eventually) up to a given number of training cycles. Afterwards, the multi-time PGD is used to decompose the time evolution in a multiscale manner, enabling the definition of a time integrator for the macrotime functions. Among other possible choices [171], the higher-order DMD is here used for the forecasting. Once

the predictor of the nonlinear term is established, it is corrected by a few high-fidelity integrations of the plasticity up to the desired final time. The linearized problem is then solved efficiently using again the multi-time PGD.

The performed numerical tests have shown that the data-driven based solver (Figure 3.19) has a significant speed-up with respect to the classic one (Figure 3.15). Given the algorithm scalability, the computational time and storage gains are further noticeable when more cycles, larger domains and finer meshes are considered, making the procedure attractive in the context of fatigue analyses.

Current research is dealing with variable amplitude loading analyses [190], complex loading scenarios as encountered in seismic engineering [56] and with the integration of damage modeling. Moreover, studies are being conducted to investigate the optimal selection of the sampling points within the corrector step.

3.4 CONCLUSIONS

As an extension of the work presented in [54], this study successfully computed timeseparated solutions within the nonlinear framework of cyclic elasto-plasticity.

In the PGD-based time multiscale procedure, the time response is calculated along two distinct time scales: the micro and the macro scales. These scales are treated as independent coordinates, and the full scale was reconstructed through their tensor product. The study demonstrated physically consistent results for the elasto-plastic response under cyclic loading. Specifically, when assigning an entire cycle to the microscale, highly nonlinear patterns emerged on the fast scale, while a smooth and slow evolution was captured by the macroscale. This rendered the macroscale characterization particularly suitable for analyses with extended time horizons, such as aging and fatigue [159]. A novel data-driven modeling approach exploiting such time multiscale representation has then been proposed.

The process involved collecting the plastic strain history and building a multi-time representation based on the PGD, followed by a forecasting of the macrotime functions. The predictor for the nonlinear term was corrected through high-fidelity integrations of plasticity, and the linearized problem was efficiently solved using multi-time PGD.

Numerical tests showed a significant speed-up in the data-driven solver (Figure 3.19) compared to the conventional one (Figure 3.15). The scalability of the algorithm resulted in notable computational time and storage savings, making it attractive for fatigue analyses.

Future research will also focus on variable amplitude loading analyses, complex loading scenarios in seismic engineering, and the integration of damage modeling. Additionally, efforts will be devoted in optimizing the selection of sampling points within the corrector step.

4 PARAMETERS

Abstract

In many engineering applications and large-scale systems, building accurate parametric surrogates can be a complex and computational demanding task. This is mainly due to two reasons: (a) the high dimension of the parametric space; (b) the computational complexity of the full-system simulation (which may require multi-physics coupling and stringent requirements on the arising meshes). A typical example is encountered in automotive engineering if one is interested in optimizing many components of the body-in-white.

In this chapter, we propose and investigate a novel approach based on a parametric domaindecomposition. The spatial domain is decomposed in non-overlapping parametric macroelements, characterized by material and geometrical parameters. The original parametric problem is recast in terms of subproblems, tackled independently by means of the proper generalized decomposition (PGD) assuming parametrized interface conditions. The parametric response of the system is obtained assembling the macroelements' responses ensuring the fields continuity at the interfaces.

The chief advantage of the proposed procedure is the non-intrusiveness which makes it compatible with usual meta-modelling approaches. Any preferred technique can be employed to build the parametric sub-solution and, eventually, the parametric modeling of the interfaces skeleton. The method is validated over academic use-cases and its industrial application is a work-in-progress.

Contents

4.1	Introduction				
4.2	4.2 Materials and methods				
	4.2.1	NURBS-based geometry mapping and PGD-based parametric solutions	85		
	4.2.2	Multi-parametric modularization	87		
	4.2.3	Computational work-flow for online real-time simulations $\ .$.	92		
4.3	4.3 Results and discussion				
	4.3.1	Steady state heat conduction	93		
	4.3.2	First order plate deformation	96		
4.4	Con	clusions	98		

4.1 INTRODUCTION

Within the framework of reduced-order models (ROMs), numerous intrusive and nonintrusive techniques have been developed to solve parametrized partial differential equations (pPDEs), for a large variety of engineering applications [191–193].

Common techniques are the snapshots-based parametric ROMs (pROMs), which rely on an offline stage where the parametric space is explored computing high-fidelity solutions of the PDE for sampled combinations of parameters values, called sampling points. This stage basically consists of exploring the so-called solution manifold for extracting a reduced approximation basis.

Within the context of the reduced basis method (RBM) [194–197], the online stage consists of projecting the solution of the full-order model (FOM) over the previously extracted reduced basis and solving the reduced problem.

Other approaches proceed by directly interpolating among the sampled snapshots, extracted orthogonal bases or subspaces [198–202]. This accelerates and simplifies the procedure, at the expense of loosing accuracy since no reduced problem is solved during the online stage. In fact, in many cases, to ensure robustness with respect to parameters variations, the interpolation must be performed on the solution manifold [203–206]. Otherwise, recent studies suggest new interpolation strategies, such as parametric optimal transport [207].

Another family of approaches is the one coming from the proper generalized decomposition (PGD) [29, 34], where a pPDE is solved accounting for the parameters as extra-coordinates, additionally to usual space and time variables. The offline stage, in this context, consists of solving a high-dimensional problem exploiting the separation of variables and defining a fixed-point based iterative algorithm. Recent studies combine the PGD-based parametric solver with NURBS-based geometrical descriptions, allowing to solve efficiently geometrically parametrized PDEs [58, 59]. As a main disadvantage, due to its intrusiveness, the PGD procedure often requires ad hoc implementations which can be difficult in case of large-scale problems or in industrial settings.

All the previously mentioned works share common issues when dealing with large and complex systems, where a single simulation can be excessively expensive computationally. Here, the curse of parameter dimensionality is encountered since such systems often exhibit a high number of parameters, compromising a rich exploration of the parametric space.

To overcome, or at least alleviate, such drawbacks, localized model reduction methods have been proposed, where standard model reduction is combined with multiscale (MS) or domain decomposition (DD) techniques [192, 208]. The primary concept of localized model reduction is the decomposition of the computational domain in modules and the definition of local reduced models, which are coupled across interfaces.

A first example is the usage of finite element tearing and interconnecting (FETI) [209–211], where the computational domain is divided into smaller subdomains or substructures, and then interconnected using interface degrees of freedom. The compatibility is enforced efficiently regardless of the differences in meshing strategies, making the method suitable for large-scale structures composed of multiple components or materials.

In the context of RBM, one can refer to [212–214], where authors target manyparameter thermo-mechanical analyses over repeated components systems. Moreover, in [215], the static condensation reduced basis method has recently been applied to efficiently model parametric wind turbines. The method has also been applied in [216] to model general cellular structures. In [217,218], several ROMs based on the proper orthogonal decomposition (POD) have been proposed for fluid-dynamics and neutron diffusion problems. Other studies successfully combine domain partitioning strategies with projection-based ROMs and hyper-reduction approaches in nonlinear and chaotic fluid-dynamics settings [219–222].

Recent works suggest nonlinear manifold ROMs based on modules modeling via neural-networks, sparse autoencoders and hyper-reduction [223]. Also in the framework of physics-informed neural networks, domain decomposition is introduced to tackle large-scale problems [224].

In some works, the interface coupling benefits of standard algorithms developed in the literature of DD. For instance, in [225] the Schwarz alternating method is used to enable ROM-FOM and ROM-ROM coupling in nonlinear solid mechanics. Similarly, in [226] the one-shot overlapping Schwarz approach is applied to component-based MOR of steady nonlinear PDEs. In [227, 228] the transmission problem along the interface is formulated in terms of a Lagrange multiplier representing the interface flux and solved through a dual Schur complement.

In the context of the proper generalized decomposition, subdomain approaches have been proposed in [70–72, 229]. In [229] a multi-patch NURBS-PGD approach has been developed with the aim of enabling or simplifying the PGD solution of problems defined over complex domains. In [70], the Arlequin method constructs local PGD solutions and uses Lagrange multipliers in overlapping regions to connect the local surrogates. A non-overlapping Dirichlet–Dirichlet method is instead used in [71], where the local surrogates are computed in the offline phase, while an interface problem is solved online to ensure the coupling. In [72], a DD-PGD framework is introduced for linear elliptic PDEs, utilizing an overlapping Schwarz algorithm to connect local surrogate models exhibiting parametric Dirichlet boundary conditions.

In this work, a general component-based pMOR framework (valuable for intrusive and non-intrusive surrogates) is presented. This is based on decomposing the spatial region in non-overlapping parametric patches. Single-patch parametric solutions are built exploiting, without loss of generality, the NURBS-PGD technique (other surrogate modeling choices could be done over a patch, also snapshots-based such as PODI and sPGD [199–201]), accounting additionally for parametric boundary conditions, as well as other parameters related to loading or physics. The full-system parametric solution is then built ensuring the equilibrium of patches across the interfaces.

The chapter is structured as follows. Section 4.2 describes all the steps of the proposed procedure: (1) definition of the interfaces skeleton; (2) identification of local model parameters; (3) identification od transmission conditions; (4) construction of local ROM; (5) imposition of the interface skeleton equilibrium. Section 4.3 shows applications two benchmark problems. The first example concerns a steady state conduction problem. The second example deals with the mechanical state of a thin elastic plate. Section 4.3 gives conclusive remarks and perspectives.

4.2 MATERIALS AND METHODS

Let us consider a parametric problem \mathcal{P} (i.e., a pPDE) defined over the physical space $\Omega \subset \mathbb{R}^d$, with d = 2, 3, and involving N_p parameters collected in the vector \boldsymbol{p} . Moreover, problem \mathcal{P} is equipped of suitable boundary conditions on the boundary of the domain $\partial\Omega$.

Let us suppose that $\Omega \subset \mathbb{R}^2$ (the physical coordinates are denoted with $\boldsymbol{x} = (x_1, x_2)$) is decomposable in a number of non-overlapping parametric parts, called modules or macroelements (each part may have its own model and/or geometric parameters). In the illustration in figure 4.1, without loss of generality, three parts have been considered. This means that Ω is expressed as $\Omega = \bigcup_{i=1}^3 \Omega_i$ and the modules intersect only on their interface, that means $\Omega_i \cap \Omega_j = \gamma_i$ or $\Omega_i \cap \Omega_j = \emptyset$ based on whether or not the modules are contiguous, as illustrated in figure 4.1. For instance, for the illustrated example, only two interfaces γ_1 and γ_2 exist.



FIGURE 4.1: Modularisation of the parametric problem \mathcal{P} defined on domain Ω and parameter vector \boldsymbol{p} into a set of some pre-solved sub-problems \mathcal{P}_i defined on Ω_i considering parameter vector \boldsymbol{p}_i .

The initial problem \mathcal{P} involving N_p parameters is recast in terms of three subproblems \mathcal{P}_i defined over the modules Ω_i and involving $N_{p,i} < N_p$ parameters, respectively.

Many possible methods have been discussed in the introduction to solve the parametric sub-problems. In this work, the NURBS-PGD approach [59] will be adopted. This mostly consists of two steps: (a) a NURBS-based geometry mapping from Ω to the reference square $\overline{\Omega}$ (the reference coordinates are denoted with $\boldsymbol{\xi} = (\xi_1, \xi_2)$) and (b) the PGD-based computation of a parametric solution $u_i^h(\boldsymbol{\xi}, \boldsymbol{p}_i)$ (and, consequently, $u_i^h(\boldsymbol{x}, \boldsymbol{p}_i)$) related to the local problem \mathcal{P}_i , characterized by the parameters collected into the vector \boldsymbol{p}_i . The two steps are schematically summarized in figure 4.2 and will be explained in detail in subsection 4.2.1.



FIGURE 4.2: Using the NURBS-PGD technique to map the parametric modules (subproblems) into a regular (hyper-cubic) computational space and then solving it.

A less direct task is the reconstruction of the parametric solution $u^h(\boldsymbol{x}, \boldsymbol{p})$ of \mathcal{P} starting from the local parametric solutions $u_i^h(\boldsymbol{x}, \boldsymbol{p}_i)$ of \mathcal{P}_i . Indeed, the parametric solutions on the different modules Ω_i must be assembled across the internal interfaces to reconstruct the response over the whole domain Ω .

The modules Ω_i have internal boundaries defined from the common interfaces as $\Gamma_1 = \gamma_1$, $\Gamma_2 = \gamma_1 \cup \gamma_2$ and $\Gamma_3 = \gamma_2$, respectively. Each subproblem \mathcal{P}_i inherits from \mathcal{P} the equations and the imposed boundary conditions over $\partial \Omega \setminus \Gamma_i$. Moreover, each \mathcal{P}_i needs to be equipped of suitable interface conditions (or transmission conditions) over Γ_i in order to satisfy the global problem \mathcal{P} .

In the parametric context, the interface conditions must be taken into account within the parametric sub-models. Indeed, the global solution is obtained by the particularization of local solutions at parameters' values, followed by the interfaces equilibrium. To tackle this point, the PGD has the chief advantage to solve BVPs with parametric boundary conditions, treating them as problem extra-coordinates. Following this rationale, the local parametric solutions will be expressed as $u_i^h(\boldsymbol{x}, \boldsymbol{p}_i, \boldsymbol{\alpha}_i)$, where the vector $\boldsymbol{\alpha}_i$ accounts for the boundary conditions imposed over the patch Ω_i , namely the interface conditions.

As an additional remark, in case of modules sharing topology and parameters, a single parametric sub-model is built and replicated in the assembly of the full system. For instance, as shown in figure 4.3, the module Ω'' can be obtained from Ω' defining local frames and opportune translations, rotations and reflections. This is what occurs, for instance, in the case of Ω_1 and Ω_3 , which are modeled through a single reference module.



FIGURE 4.3: Two modules sharing topology and parameters.

All the details about the parametric modularization and assembly proposed in this work will be given in subsection 4.2.2.

4.2.1 NURBS-based geometry mapping and PGD-based parametric solutions

Denoting with \boldsymbol{p} a finite-dimensional vector of parameters (these can be model/material parameters \boldsymbol{p}^m and/or geometrical parameters \boldsymbol{p}^g), the abstract discrete weak formulation of a generic pPDE over the domain $\Omega(\boldsymbol{p}^g)$ reads:

Find
$$u_{\mathbf{p}}^{h} \in \mathcal{U}_{\mathbf{p}}^{h}$$
 such that $a_{\mathbf{p}}(u_{\mathbf{p}}^{h}, v^{h}) = \ell_{\mathbf{p}}(v^{h}) \quad \forall v^{h} \in \mathcal{U}_{\mathbf{p}}^{h},$ (4.2.1)

with \mathcal{U}_p^h abstract discrete functional space and $a_p(\cdot, \cdot)$, $\ell_p(\cdot)$ parametrized bilinear and linear forms, respectively.

The parametric problem (4.2.1) can be efficiently solved by means of the PGD, where a unique computation is conducted in high-dimension treating the parameters as extra coordinates.

Our recent works investigate the usage of parametric NURBS-based geometry descriptions with the PGD framework [59]. In this context, a mapping from the reference square domain $\overline{\Omega}$ (the reference coordinates are denoted with $\boldsymbol{\xi}$) to the physical domain $\Omega(\boldsymbol{p}^g)$ (the physical coordinates are denoted with \boldsymbol{x}) is exploited to define the parameter-dependent PDE over the fixed domain $\overline{\Omega}$:

$$F_{p^g} \colon \bar{\Omega} \to \Omega(p^g)$$

$$\boldsymbol{\xi} \mapsto \boldsymbol{x} = F_{p^g}(\boldsymbol{\xi}) = \sum_{i=1}^n \sum_{j=1}^m R_{ij}^{pq}(\boldsymbol{\xi}) \boldsymbol{B}_{ij}(p^g), \qquad (4.2.2)$$

where **B** refers to the control points (the vertices of the so-called control net) in the physical domain and $R_{ij}^{pq}(\xi,\eta)$ denote the bivariate NURBS basis functions in 2D domains whose expressions are detailed in appendix B.1.

In particular, the dependency of the domain upon the parameters p^{g} is expressed through the control points of the NURBS surface describing the domain. A schematic representation of domain geometry and the control net is shown in figure 4.4. In this figure, the control net that defines the NURBS object is shown using dash lines and the control points are shown using solid dots. Moreover, the domain is parametrized through one of its control points. Changing the value of p, the related shapes $\Omega_{x}(p)$ are generated, as shown in the right-side of figure 4.4.



FIGURE 4.4: Physical domain parametrized by one of its control points (white dot) and several shapes obtained by moving its location.

Once the mapping (4.2.2) is introduced, the weak formulation (4.2.1) can be mapped back to the fixed domain $\overline{\Omega}$:

Find
$$\bar{u}^h_p \in \bar{\mathcal{U}}^h$$
 such that $\bar{a}_p(\bar{u}^h_p, \bar{v}^h) = \bar{\ell}_p(\bar{v}^h) \quad \forall \bar{v}^h \in \bar{\mathcal{U}}^h,$ (4.2.3)

where quantities $\overline{\cdot}$ have opportunely been mapped into the reference domain.

Remark 4.2.1. Notice that when mapping the differential operators from the physical domain to the reference one, to obtain problem 4.2.3, one must account for the Jacobian of the transformation (4.2.2). This aspect is discussed in appendix B.2.

At this point, the dependency on problem parameters $\boldsymbol{p} = (\boldsymbol{p}^m, \boldsymbol{p}^g)$ is separated from the one on space variables $\boldsymbol{\xi}$. Formulation (4.2.3) is integrated over the parametric domain $\Omega_{\boldsymbol{p}}$ and rewritten in a tensorial formalism:

Find
$$\bar{u}^h \in \bar{\mathcal{U}}^h \otimes L^2(\Omega_p)$$
 such that $\bar{A}(\bar{u}^h, \bar{v}^h) = \bar{L}(\bar{v}^h) \quad \forall \bar{v}^h \in \bar{\mathcal{U}}^h,$ (4.2.4)

where

$$\bar{A}(\bar{u}^h, \bar{v}^h) = \int_{\Omega_p} \bar{a}_p(\bar{u}^h, \bar{v}^h) \mathrm{d}p, \quad \bar{L}(\bar{v}^h) = \int_{\Omega_p} \bar{\ell}_p(\bar{v}^h) \mathrm{d}p.$$

The solution $\bar{u}_{\boldsymbol{p}}^h$ is sought into the low-rank separated form

$$\bar{u}^{h}(\boldsymbol{\xi}, \boldsymbol{p}) \approx \bar{u}^{h}_{m}(\boldsymbol{\xi}, \boldsymbol{p}) = \sum_{i=1}^{m} \bar{\psi}^{h}_{i}(\boldsymbol{\xi}) G_{i}(\boldsymbol{p}), \qquad (4.2.5)$$

where $m \in \mathbb{N}^*$ denotes the PGD rank (number of modes), $\bar{\psi}_i^h \in \bar{\mathcal{U}}^h$ the physical space modes and $G_i(\mathbf{p}) \in L^2(\Omega_{\mathbf{p}})$ the parametric modes.

Remark 4.2.2. Comparing the NURBS-based geometry mapping (4.2.2) and the PGDbased solution (4.2.5), it becomes clear the interest of the NURBS-PGD technique. Indeed, as one can observe from (4.2.2), the contribution of the geometric parameters p^{g} is, by definition, separated from the one of the reference space coordinates $\boldsymbol{\xi}$. Such separation is exactly the same enforced in the PGD solution 4.2.5.

Moreover, the separability of the reference domain $\overline{\Omega}$ and of the parametric space Ω_p can be exploited to achieve a fully separated representation of the solution $\overline{u}^h(\boldsymbol{\xi}, \boldsymbol{p})$.

Considering $\mathbf{p} = (p_1, \ldots, p_{N_p})$ the vector gathering the parametric extra-coordinates, the parameter-separated PGD solution reads:

$$\bar{u}^{h}(\boldsymbol{\xi}, \boldsymbol{p}) \approx \bar{u}^{h}_{m}(\boldsymbol{\xi}, \boldsymbol{p}) = \sum_{i=1}^{m} \bar{\psi}^{h}_{i}(\boldsymbol{\xi}) \prod_{l=1}^{N_{p}} G_{i,l}(p_{l}).$$
(4.2.6)

Following the same rationale, the corresponding discrete functional spaces is decomposed as $L^2(\Omega_p) = \bigotimes_{l=1}^{N_p} L^2(\Omega_{p_l})$, leading to a slightly modified variant of the weak formulation (4.2.4).

The solution expressed in (4.2.6) is computed by means of an iterative algorithm which alternates among the different axes ξ_i and p_i , leading to a sequence of one-dimensional separated problems. Given the reduced computational cost, really fine grids can be used over such axes, allowing to obtain rich solutions also in case of high-dimensional parametric spaces.

For instance, figure 4.5 shows the separated structure of the PGD computational domain, in the case of 3 parameters, that is $N_p = 3$. The extension of the mapped physical space $\Omega_{\boldsymbol{\xi}}$ by the geometry parameter space $\Omega_{\boldsymbol{p}}$ is a five dimensional computational space $\Omega \subset \mathbb{R}^5$. Therefore, any point $\boldsymbol{\vartheta} \in \Omega$ is defined as $\boldsymbol{\vartheta} = (\vartheta_1, \vartheta_2, \vartheta_3, \vartheta_4, \vartheta_5) = (\xi_1, \xi_2, p_1, p_2, p_3).$

Problem (4.2.4) is therefore solved in the computational domain Ω and the PGD solution can thus be particularized for any choice of the shape parameters defining Ω_p . Let us consider, for instance, the parametric domain illustrated in figure 4.4, whose shape depends on the location of the white dot (parameter p). Solving a generic BVP over this domain in the sense of problem (4.2.4), the obtained solution can be evaluated in real time for any possible value of the parameter p, as shown in figure 4.6.

4.2.2 Multi-parametric modularization

Let us consider the general case of a parametric problem \mathcal{P} rewritten in terms of P parametric subproblems \mathcal{P}_p .

The methodology can be summarized with the following steps:



FIGURE 4.5: PGD computational domain after separation of space and parameters.



FIGURE 4.6: Example of PGD solution over a parametrized geometry.

- 1. define the interfaces skeleton by modules decomposition;
- 2. identify the model parameters of each module;
- 3. assume a reduced model of the transmission conditions (Neumann or Dirichlet);
- 4. find a parametric solution for each module, including the interface conditions and the model/geometrical parameters;
- 5. find the parameters ensuring compatibility among the parts sharing the interface.

These steps will be explained in detail in the subsections here below.

INTERFACES SKELETON The domain Ω is decomposed in P non-overlapping modules Ω_p and in S internal interfaces γ_s , $s = 1, \ldots, S$, linking the different components and characterizing the structure skeleton Γ . This is sketched in figure 4.7.

Let us denote with S_p the set of indices associated with the internal interfaces characterizing Ω_p . This means that $\partial \Omega_p = \Gamma_p \cup \bigcup_{s \in S_p} \gamma_s$, where $\Gamma_p = \partial \Omega \cap \partial \Omega_p$ is the external part of the boundary.

Moreover, let \mathcal{V}_s the set of indices of modules sharing the interface γ_s . That is, if $\gamma_s = \partial \Omega_l \cap \partial \Omega_m$ then $\mathcal{V}_s = \{l, m\}$. Without loss of generality, this is sketched in figure 4.8 considering modules as squares. However, as explained previously, any complex shape can be mapped into the square using single-patch or multi-patch NURBS.



FIGURE 4.7: Large structure composed by P parts linked by S interfaces.



FIGURE 4.8: Two parts sharing the same interface γ_s .

SINGLE PART PARAMETERS Each part composing the entire structure has its own parameters, collected into the vector p_p . Figure 4.9 shows how the structure is decomposed into independent parametric components.



FIGURE 4.9: Parametric sub-components.

REDUCED SKELETON KINEMATICS The parametric single-part response u_p^h is the parametric solution of the problem \mathcal{P}_p equipped of suitable boundary conditions imposed over $\partial\Omega_p$, this means $u_p^h(\boldsymbol{x}, \boldsymbol{p}_p, u|_{\partial\Omega_p})$.

Let us suppose that such boundary condition over $\partial \Omega_p$ can be characterized through a set of coefficients (parameters) $\boldsymbol{\alpha}_s$. Then, for a fixed choice of parameters \boldsymbol{p} , the solution of problem \mathcal{P} can be obtained evaluating the solutions of the subproblems \mathcal{P}_s satisfying some compatibility conditions among the parameters $\boldsymbol{\alpha}_s$ at the internal interfaces.

In this framework, coefficients $\boldsymbol{\alpha}_s$ represent a reduced model of the interface behavior depending on the chosen parametrization for the boundary conditions. For instance, let us suppose that R coefficients are enough to have a good parametrization, $\boldsymbol{\alpha}_s = (\alpha_s^1, \ldots, \alpha_s^R)$. Consequently, the interface field in the whole skeleton Γ can be represented by the weights associated with all the interfaces, yielding the reduced model $\boldsymbol{\Lambda} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_S) \in \mathbb{R}^{R \times S}$. *Remark* 4.2.3. The parametrization of the interface conditions increases the dimensionality of the local parametric problem, introducing new dimensions related to the coefficients, representing a limit of the procedure if many parameters are needed. In [72], the issue is faced splitting the local problem (exploiting the linearity) into a family of subproblems involving sufficiently small sets of active boundary parameters.

This is illustrated in figure 4.10.



FIGURE 4.10: Parametric sub-components with parametrized boundary condition.

Remark 4.1. Such reduced model of the interface can also be physics-informed. In this sense, a few high-fidelity solutions of the entire problem could be simulated for a series of loading conditions. From each of these simulations the field at the structure skeleton can be extracted. Let us denote with $u^i|_{\gamma_s} = (u^i, v^i, w^i)|_{\gamma_s}$ the displacement coming from the *i*-th high-fidelity simulation, extracted at the interface γ_s . A reduced kinematic can be constructed by applying the POD at the interface γ_s . The weights associated with the first R modes are grouped into the vector $\boldsymbol{\alpha}_s = (\alpha_s^1, \ldots, \alpha_s^R)$. Consequently, the displacement in the whole skeleton Γ can be represented by the weights associated with all the interfaces, yielding the reduced model $\boldsymbol{\Lambda} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_S) \in \mathbb{R}^{R \times S}$.

PART REDUCED MODEL: PARAMETRIC TRANSFER FUNCTION Each subproblem \mathcal{P}_p can be solved using the NURBS-PGD method accounting for parameters p_p and for the reduced skeleton kinematics Λ as additional coordinates. In particular, the parametric transfer function of each part involves thus the internal parameters p_p and the coefficients parametrizing the kinematics at its boundaries, that is α_s with $s \in \mathcal{S}_p$.

The parametric transfer function allows then the computation of the related thermomechanical fields, like the temperature/displacement contour

$$\boldsymbol{u}|_{p} = \boldsymbol{u}|_{p}(\boldsymbol{p}_{p}, \boldsymbol{\alpha}_{s}), \quad \forall s \in \mathcal{S}_{p}.$$
 (4.2.7)

In particular, the fluxes/forces at the interface γ_s characterizing the part \mathcal{P}_p can be extracted

$$oldsymbol{F}|_s^p = oldsymbol{F}|_s^p(oldsymbol{p}_p,oldsymbol{lpha}_s), \quad orall s \in \mathcal{S}_p.$$

ENFORCING THE INTERFACE EQUILIBRIUM The skeleton kinematics $\boldsymbol{u}|_{\Gamma}$ is univocally determined by its reduced representation $\boldsymbol{\Lambda}$. For a new choice of parameters \boldsymbol{p}_p , the compatibility of interface fields must be enforced to determine the correct kinematics $\boldsymbol{\Lambda}^*$. This amounts to minimize a given cost function $\mathcal{C}|_{\Gamma}$ which depends upon the problem at hand.

In structural mechanics, an example of cost function $C|_{\Gamma}$ can be the sum of forces (and momentums if needed) at all the interfaces γ_s (in a thermal problem, this can be the equilibrium of fluxes), that is

$$\sum_{s=1}^{S} \mathcal{C}|_{\gamma_s} = \sum_{s=1}^{S} \sum_{p \in \mathcal{V}_s} \boldsymbol{F}|_s^p = \sum_{s=1}^{S} \sum_{p \in \mathcal{V}_s} \boldsymbol{F}|_s^p (\boldsymbol{p}_p, \boldsymbol{\alpha}_s).$$

For instance, considering two parts \mathcal{P}_l and \mathcal{P}_m sharing the interface γ_s , this stage consists of determining the coefficients $\boldsymbol{\alpha}^*$ minimizing the cost function \mathcal{C} at the interface γ_s . This is illustrated in figure 4.11.



FIGURE 4.11: Compatibility of interface between two parts.

ITERATIVE SCHEME For an arbitrary choice of the skeleton kinematics $\mathbf{\Lambda}_0 = (\boldsymbol{\alpha}_1^0, \ldots, \boldsymbol{\alpha}_S^0)$, certainly the equilibrium is not satisfied at all the interfaces γ_s . Thus, we assume the existence of at least one interface violating the equilibrium, e.g. γ_s

$$\sum_{p \in \mathcal{V}_s} \mathbf{F}|_s^p(\boldsymbol{p}_p, \boldsymbol{\Lambda}_0) \neq \mathbf{0}.$$
(4.2.8)

In what follows the unbalanced forces at each interface will be noted \mathbf{R}_s ,

$$\mathbf{R}_{s} = \sum_{p \in \mathcal{V}_{s}} \mathbf{F}|_{s}^{p}(\boldsymbol{p}_{p}, \boldsymbol{\Lambda}_{0}), \quad \forall s = 1, \dots, S.$$
(4.2.9)

In order to equilibrate the system one should modify the skeleton kinematics, $\Lambda_0 + \Delta \Lambda$ (that is, $\alpha_s^0 + \Delta \alpha_s$, $\forall s = 1, ..., S$), in order to satisfy the equilibrium everywhere, we should enforce

$$\sum_{p \in \mathcal{V}_s} \mathbf{F}|_s^p(\boldsymbol{p}_p, \boldsymbol{\Lambda}_0 + \Delta \boldsymbol{\Lambda}) = \mathbf{0}, \quad \forall s = 1, \dots, S$$
(4.2.10)

with

$$\mathbf{F}|_{s}^{p}(\boldsymbol{p}_{p},\boldsymbol{\Lambda}_{0}+\Delta\boldsymbol{\Lambda})\approx\mathbf{F}|_{s}^{p}(\boldsymbol{p}_{p},\boldsymbol{\Lambda}_{0})+\left.\frac{\partial\mathbf{F}|_{s}^{p}(\boldsymbol{p}_{p},\boldsymbol{\Lambda})}{\partial\boldsymbol{\Lambda}}\right|_{\boldsymbol{\Lambda}_{0}}\Delta\boldsymbol{\Lambda},$$
(4.2.11)

leading to the Newton-Raphson iterate

$$\sum_{p \in \mathcal{V}_s} \frac{\partial \mathbf{F}|_s^p(\boldsymbol{p}_p, \boldsymbol{\Lambda})}{\partial \boldsymbol{\Lambda}} \bigg|_{\boldsymbol{\Lambda}_0} \Delta \boldsymbol{\Lambda} = -\mathbf{R}_s, \qquad (4.2.12)$$

that can be assembled in a linear system

$$\begin{pmatrix} \mathbf{M}_{11}(\boldsymbol{p}_{11}) & \mathbf{M}_{11}(\boldsymbol{p}_{12}) & \cdots & \mathbf{M}_{1S}(\boldsymbol{p}_{1S}) \\ \mathbf{M}_{21}(\boldsymbol{p}_{21}) & \mathbf{M}_{21}(\boldsymbol{p}_{22}) & \cdots & \mathbf{M}_{2S}(\boldsymbol{p}_{2S}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}_{S1}(\boldsymbol{p}_{S1}) & \mathbf{M}_{S1}(\boldsymbol{p}_{S2}) & \cdots & \mathbf{M}_{SS}(\boldsymbol{p}_{SS}) \end{pmatrix} \begin{pmatrix} \Delta \boldsymbol{\alpha}_1 \\ \Delta \boldsymbol{\alpha}_2 \\ \vdots \\ \Delta \boldsymbol{\alpha}_S \end{pmatrix} = - \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \vdots \\ \mathbf{R}_S \end{pmatrix}, \quad (4.2.13)$$

where \mathbf{M}_{ij} contains the contributions of interface γ_j on interface γ_i , and consequently the parameters \mathbf{p}_{ij} involved are the ones related to the part that involves interfaces γ_i and γ_j , with $\mathbf{M}_{ij} = \mathbf{0}$ if no part contains both interfaces γ_i and γ_j .

Remark 4.2. System (4.2.13) only involves few hundred of equations and consequently can be solved extremely fast. However, in case of many interfaces, its assembly requires the evaluation of the cost function and computation of the gradient (with respect to the parameters), in an iterative setting. Thus, if solved online, the real-time response in some scenarios could be compromised (this is the case for any optimization in high dimension).

A valuable route consists of solving thousands of times (offline) the system (4.2.13) for a diversity of choices of parameters $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_P$, for computing the associated kinematics $\boldsymbol{\Lambda}(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_P)$. For that purpose powerful regression techniques could be employed, e.g. neural networks-based deep learning.

Another valuable route could be representing the different values of Λ_k , related to the parameter choice p_1^k, \ldots, p_P^k , to check its intrinsic dimensionality, by employing for example manifold learning (such as the kPCA [230] or LLE [231], for instance) or even auto-encoders. The main interest of such a reduction is the possibility of employing standard regressions, such as the sPGD [199,200].

Remark 4.3. The methodology requires to increase the number of parameters since the boundary conditions of single patches are parametric. However, this does not face the curse of dimensionality contrarily to full-structure based modeling. Indeed, considering a structure composed by P parts each one involving N_p parameters, the total number of parameters in the problem is PN_p . Le us split the full problem in P subproblems involving N_p parameters each and R additional parameters for the interface conditions. Then, the total number of parameters in each local problem is $N_p + R$. Even in the case in which $N_p + R$ is comparable with PN_p , the methodology is convenient since all local models are all built in parallel over simple geometries. In target applications involving many parts and many parameters parameters by parts $N_p + R \ll PN_p$ and $R \ll N_p$ ensuring the good scalability of the algorithm.

4.2.3 Computational work-flow for online real-time simulations

The global proposed workflow consists of:

- 1. offline stage:
 - (a) structure decomposition in a number of modules and determination of the interfaces;
 - (b) creation of a reduced model of the interface conditions: this can be achieved via an a priori parametrization; otherwise, one can compute some highfidelity solutions of \$\mathcal{P}\$ for some parameters' combinations and determine the principal modes on the skeleton composed by the interfaces;
 - (c) construction of the parametric solutions for each problem \mathcal{P}_p , for $p = 1, \ldots, P$ (this step is performed efficiently in parallel);
- 2. online stage:
 - (a) choosing the model parameters $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_P$;

- (b) use the part parametric transfer function (4.2.7), $\boldsymbol{u}|_p = \boldsymbol{u}|_p(\boldsymbol{p}_p, \boldsymbol{\alpha}_s)$ for each module;
- (c) find the interface parameters ensuring the global equilibrium of modules.

4.3 **Results and Discussion**

4.3.1 Steady state heat conduction

A first example is a design problem in 2D steady state conduction. The domain is a curved corner L-shaped geometry having 6 shape parameters $\boldsymbol{p} = (p_1, \ldots, p_6)$, as illustrated in figure 4.12. Moreover the inlet and outlet fluxes q_1 and q_2 are parametric and described by 3 coefficients, that is $\boldsymbol{\beta}^{\text{in}} = (\beta_1, \beta_2, \beta_3)$ and $\boldsymbol{\beta}^{\text{out}} = (\beta_4, \beta_5, \beta_6)$, respectively. Null flux and fixed temperature are considered as boundary conditions for the outer and inner wall, respectively. Denoting with $\boldsymbol{\beta} = (\boldsymbol{\beta}^{\text{in}}, \boldsymbol{\beta}^{\text{out}})$ the vector collecting the parameters related to the boundary conditions, the sought parametric solution has 12 parameters and can be written as $u(x, y, \boldsymbol{p}, \boldsymbol{\beta})$.



FIGURE 4.12: Steady state conduction problem set-up.

The problem is decomposed in three parametric sub-problems \mathcal{P}_i defined over Ω_i , for $i = 1, \ldots, 3$, having the corresponding geometrical parameters p_i . Moreover, the internal interfaces γ_1 and γ_2 are treated considering a parametric flux profile still described with 3 parameters, as shown in figure 4.13. For instance, the interface condition between the domain Ω_i and Ω_j is an imposed flux depending upon the parameters $\boldsymbol{\alpha}^{ij} = (\alpha_1^{ij}, \alpha_2^{ij}, \alpha_3^{ij})$.

Since problems \mathcal{P}_1 and \mathcal{P}_3 exhibit exactly the same parameters' dependency, they can be reduced to a single parametric problem $\tilde{\mathcal{P}}$ as shown in figure 4.14, where both inlet and outlet fluxes are parametric. The angle φ_k represents the rotation of the domain to be considered for the assembly within the global system.

Letting $\tilde{\boldsymbol{p}} = (\tilde{p}_1, \tilde{p}_2, \tilde{p}_3)$ the geometrical parameters and $\tilde{\boldsymbol{\beta}}$ the ones related to the parametric flux, the sought solution of $\tilde{\mathcal{P}}$ is $\tilde{u}(x, y, \tilde{\boldsymbol{p}}, \tilde{\boldsymbol{\beta}})$ with 9 parameters. The solution u_i of \mathcal{P}_i , for $i = \{1, 3\}$, is simply the particularization of \tilde{u} at the correct



FIGURE 4.13: Parametric patches decomposition.

parameters values, that is $u_i = \tilde{u}(x, y, \boldsymbol{p}_i, \boldsymbol{\beta}_i)$, where

$$\begin{cases} p_1 = (a_1, c_1, b_1), \\ \beta_1 = (\beta^{\text{in}}, \alpha^{12}), \end{cases} \text{ and } \begin{cases} p_3 = (a_3, b_3, c_3), \\ \beta_3 = (\alpha^{23}, \beta^{\text{out}}). \end{cases}$$

In the same way, the solution of problem \mathcal{P}_2 has 2 geometrical parameters $\mathbf{p}_2 = (a_2, b_2)$ and 6 parameters for the inlet and outlet fluxes, that is $\boldsymbol{\alpha}^{12}$ and $\boldsymbol{\alpha}^{23}$, respectively.

$$\begin{array}{c} \tilde{\mathcal{P}} & q = 0 \\ q^{\text{in}}(\tilde{\beta}_1, \tilde{\beta}_2, \tilde{\beta}_3) & y_k & \tilde{\Omega} \\ y_k & \tilde{\Omega} \\ y_k & \chi_k \\ y_k & \chi_k \\ y_k & \chi_k \\ y_k & \chi_k \\ u = 0 \end{array} q^{\text{out}}(\tilde{\beta}_4, \tilde{\beta}_5, \tilde{\beta}_6)$$

FIGURE 4.14: Reference sub-problem for \mathcal{P}_1 and \mathcal{P}_3 .

The three parametric sub-solutions finally read

$$\begin{cases} u_1(x, y, \boldsymbol{p}_1, \boldsymbol{\beta}^{\text{in}}, \boldsymbol{\alpha}^{12}), \\ u_2(x, y, \boldsymbol{p}_2, \boldsymbol{\alpha}^{12}, \boldsymbol{\alpha}^{23}), \\ u_3(x, y, \boldsymbol{p}_3, \boldsymbol{\alpha}^{23}, \boldsymbol{\beta}^{\text{out}}), \end{cases}$$
(4.3.1)

and the global solution $u(x, y, \boldsymbol{p}, \boldsymbol{\beta})$ is obtained by assembly. To this purpose, one must ensure the geometric constraint fixing $a_1 = b_2 = c_3 = p_4$ and, for a given choice of parameters, minimize the temperature jumps at the interface, by solving

$$\min_{\alpha^{12},\alpha^{23}} (\|u_1 - u_2\|_{2,\gamma_1} + \|u_2 - u_3\|_{2,\gamma_2}),$$
(4.3.2)

where $\|\cdot\|_{2,\gamma_i}$ denotes the Euclidean norm at the interface γ_i .

In this example, the geometrical parameters are chosen varying in [1,3] while all the coefficients of flux profiles have range [-100, 100]. The metamodels \tilde{u} (i.e., u_1 and u_3) and u_2 are computed in parallel, employing the NURBS-PGD method. Each direction (space and parameters ones) is discretized in 51 nodes, yielding a total number of DOFs of 51¹¹ and 51¹⁰ for $\tilde{\mathcal{P}}$ and \mathcal{P}_2 , respectively. This is possible thanks to the usage of PGD-based separated representations. A simple hill climbing algorithm is employed for the minimization step.

Figure 4.15 shows 12 snapshots of the sub-solutions for different combinations of the parameters. This can be seen as a catalog of patches which can suitably be assembled to evaluate several designs of the curved L-shape geometry of figure 4.12, as shown in figure 4.16.



FIGURE 4.15: 12 snapshots of parametric sub-solutions.



FIGURE 4.16: 6 snapshots of the assembled parametric solution.

4.3.2 FIRST ORDER PLATE DEFORMATION

Let us consider a thin elastic plate composed of two parts with different geometric and material properties, as shown in figure 4.17. The plate is clumped along the bottom edge (homogeneous Dirichlet boundary condition), while along the left and right edges a distributed out-of-plane loading is applied. The upper edge has a traction-free boundary condition (homogeneous Neumann). The mechanical problem \mathcal{P} has 17 geometric and model parameters resumed in table, with also the corresponding ranges 4.1.



FIGURE 4.17: Plate problem set-up.

Length	Thickness	Young Modulus	Poisson Ratio	Force
a_1, b_1, c_1, a_2, c_2	h_1, h_2	E_1, E_2	$ u_1, u_2 $	β_1,\ldots,β_6
[0.1, 0.3]	[0.001, 0.01]	$[100, 300] \cdot 10^9$	0.2, 0.4	$[-1, -1] \cdot 10^4$

TABLE 4.1: Problem parameters and ranges.

Each out-of-plane force $F_{z,i}$, for i = 1, 2 is assumed depending upon 3 coefficients β_i . The two plates share the interface γ_1 whose length is determined by the parameter c_1 . The plate is modeled through the first order Kirchhoff theory, which expresses the displacement field in terms of the middle plane kinematic variables w, θ_x and θ_y ,

$$\begin{cases} u(x, y, z) = z\theta_y(x, y) \\ v(x, y, z) = -z\theta_x(x, y) \\ w(x, y, z) = w(x, y) \end{cases}$$
(4.3.3)

where θ_x and θ_y are the angles defining the rotation of the normal vector to the middle surface and w is the vertical displacement (deflection).

Following the same procedure, problem \mathcal{P} can be split in two sub-problems which can be reduced to the same parametric macro-element $\tilde{\Omega}$, as resumed in figure 4.18.

In this way, the original problem is reduced to a single problem $\tilde{\mathcal{P}}$ having 12 parameters, and the two particularized sub-solutions can be written as

$$\begin{cases} \boldsymbol{u}_{1}(x, y, z, \boldsymbol{p}_{1}, \boldsymbol{\beta}_{1}, \boldsymbol{\alpha}^{12}), \\ \boldsymbol{u}_{2}(x, y, z, \boldsymbol{p}_{2}, \boldsymbol{\alpha}^{12}, \boldsymbol{\beta}_{2}), \end{cases}$$
(4.3.4)

where $p_i = (a_i, b_i, c_i, h_i, E_i, \nu_i)$ for $i = \{1, 2\}$.

Finally, the assembly is performed imposing the geometrical constraint $c_1 = b_2 = c_3$ and the minimization of the displacement jump at the interface, that is

$$\min_{\alpha^{12}} \| \boldsymbol{u}_1 - \boldsymbol{u}_2 \|_{2,\gamma_1}, \tag{4.3.5}$$



FIGURE 4.18: From modules to a reference parametric patch.

where $\left\|\cdot\right\|_{2,\gamma_1}$ denotes the Euclidean norm at the interface γ_1 .

For instance, figure 4.19 shows the assembled solution when choosing the following parameters values

$$\begin{cases} \boldsymbol{p}_1 = (0.3, 0.15, 0.2, 0.01, 100 \cdot 10^9, 0.3), \ \boldsymbol{\beta}_1 = (7 \cdot 10^3, 7 \cdot 10^3, 7 \cdot 10^3), \\ \boldsymbol{p}_2 = (0.2, 0.2, 0.25, 0.01, 200 \cdot 10^9, 0.3), \ \boldsymbol{\beta}_2 = (0, -7 \cdot 10^3, -7 \cdot 10^3). \end{cases}$$
(4.3.6)



FIGURE 4.19: Plate bending from different isometric views.

4.4 CONCLUSIONS

In the presented work, a new methodology has been introduced for the construction of parametric surrogates within the context of high-dimensional problems. The foundation of this approach lies in the strategic decomposition of the domain into nonoverlapping parametric modules. The most appealing feature of the method is the simultaneous treatment of these modules, akin to the principles of parallel computing, enhancing efficiency and computational speed.

The segmentation of the domain facilitates the independent handling of each parametric module, and subsequently, accelerates the development of sub-solutions. Afterwards, an assembly process is necessary to describe the parts interactions within the global system. This is performed via a minimization technique that ensures compatibility at the interfaces of the distinct parts. Notably, the local surrogates are built via the NURBS-PGD approach. Although the NURBS-PGD method is chosen for its efficacy in addressing parametric design problems, it is important to underscore that alternative procedures could be seamlessly integrated based on specific requirements and preferences (non-intrusive character of the procedure).

One of the most compelling attributes of this methodology lies in the replicability of individual component models. This feature proves to be invaluable for expediting the design process and facilitating prompt performance evaluations. The broader applicability of this technique extends to diverse industrial and large-scale applications, a focal point of our ongoing research line. Emphasis is particularly placed on refining the minimization step, whose role is pivotal in ensuring a real-time response.

CONCLUSIONS

This thesis has explored important issues encountered in the field of simulation-based engineering for specific industrial applications. It takes a distinct approach by individually addressing the challenges posed by space, time, and parameters.

In the spatial domain, the primary concern revolves around effectively incorporating three-dimensional effects within degenerate domains, particularly in the context of plates and shells like structures.

In the time domain, the difficulties are mostly related to the complexities of performing accurate fine-scale numerical simulations for exceptionally prolonged phenomena, within the context of cyclic elastic-plastic fatigue simulations.

Regarding parameters, the focus stands in the construction of parametric meta-models for large multi-component structures and domains, as can be the case for a full-vehicle parametric modeling.

All the different topics have been addressed through separated representations within the framework of the proper generalized decomposition (PGD). In particular, the plate-like domain (3D) is separated in an in-plane (2D) and out-of-plane (1D) one. The time domain is decomposed in a multi-time manner, via two independent variables, the macro-time (slow time) and the micro-time (fast time). The parametric multicomponent latlatstructure is separated in multiple parametric subdomains.

An important requirement in the development of new proposed solutions is favoring their integration in commercial solvers with minimal intrusiveness, which constitutes the main achievement of this thesis. To this purpose, all the procedures exploit some pre-existing features of the commercial codes and enhance them through external plug-ins.

The newly developed enhanced-shell formulation for the space makes use of a slightly modified version of the currently accessible assembly of standard shell formulations, while improving the out-of-plane kinematics through the use of an externally trained machine learning framework.

In cyclic fatigue simulations, a forecasting framework is built for the macroscale evolution and the microscale response can be recomputed within the industrial solver.

A multi-component parametric structure can be decomposed in sub-components for which the parametric model can be built via already available tools.

Although the proposed solutions have been tested over benchmark cases, they open valuable perspective in industrial contexts, which are being addressed as a work in progress with ESI Group.



IMPLEMENTATION ASPECTS OF PGD, G-POD AND HODMD

A.1 TENSOR-BASED PGD METHODS

Here we shortly recall the PGD algorithm in a general tensor framework [31, 32, 35]. The solution of a multidimensional problem is built as a sum of tensor products of functions defined in some sub-spaces with moderate dimension (1, 2 or 3), thus providing a general separable representation form. In particular, the authors make use of the best rank-1 approximation property of tensors of order 3 or higher [232] (in [233], it has been proved that tensors of order 3 or higher can fail to have best rank-n approximation) to propose an iterative method based on the so-called *projection-enrichment technique* [31].

Let Ω be a multidimensional domain involving several coordinates x_i (not necessarily one-dimensional), which can coincide, for instance, with spatial coordinates, time, model or geometric parameters. We consider the weak formulation of a linear problem:

$$Find \ \psi \in V(\Omega) \ s.t.$$
$$a(\psi(x_1, \dots, x_d), \psi^*(x_1, \dots, x_d)) = b(\psi^*(x_1, \dots, x_d)) \quad \forall \psi^* \in V(\Omega),$$
(A.1.1)

with $V(\Omega)$ an adequate function space ensuring the well-posedness of such a formulation.

We assume that, after a discretization of problem (A.1.1), we are lead to solve the linear system:

$${}^{\mathrm{t}}\boldsymbol{\psi}^{*}\mathcal{A}\boldsymbol{\psi} = {}^{\mathrm{t}}\boldsymbol{\psi}^{*}\mathcal{B},\tag{A.1.2}$$

where the operator on the left-hand side and the right-hand side member are expressed in a separated form as

$$\mathcal{A} = \sum_{k=1}^{N_A} \mathbf{A}_1^k \otimes \mathbf{A}_2^k \otimes \cdots \otimes \mathbf{A}_d^k, \quad \mathcal{B} = \sum_{k=1}^{N_B} \mathbf{b}_1^k \otimes \mathbf{b}_2^k \otimes \cdots \otimes \mathbf{b}_d^k.$$
(A.1.3)

The approximated PGD solution ψ is sought in the discrete separated form

$$\boldsymbol{\Psi} = \sum_{j=1}^{N_F} \alpha_j \, \mathbf{f}_1^j \otimes \mathbf{f}_2^j \otimes \ldots \otimes \mathbf{f}_d^j, \tag{A.1.4}$$

and an analogous representation is adopted for the test function ψ^* , being

$$\mathbf{\psi}^* = \mathbf{f}_1^i \otimes \mathbf{f}_2^i \otimes \ldots \otimes \mathbf{f}_d^i,$$

with $i = 1, \ldots, N_F$.

Solution $\boldsymbol{\psi}$ is built-up by using a projection-enrichment iterative scheme [31]. In particular, it will be assumed that the global convergence is attained when the error estimator $\varepsilon = \|\mathcal{A}\boldsymbol{\psi} - \mathcal{B}\|_2$ is small enough, where $\|\cdot\|_2$ denotes the standard Frobenius norm of a tensor of order d.

The projection stage consists of finding the set of coefficients α_j in (A.1.4) verifying the relations

$$\sum_{j=1}^{N_F} H_{ij} \alpha_j = J_i \quad \forall i = 1, \dots, N_F$$
(A.1.5)

where

$$H_{ij} = \sum_{k=1}^{N_A} \left({}^{\mathrm{t}} \mathbf{f}_1^i \, \mathbf{A}_1^k \, \mathbf{f}_1^j \right) \cdot \left({}^{\mathrm{t}} \mathbf{f}_2^i \, \mathbf{A}_2^k \, \mathbf{f}_2^j \right) \cdot \ldots \cdot \left({}^{\mathrm{t}} \mathbf{f}_d^i \, \mathbf{A}_d^k \, \mathbf{f}_d^j \right),$$
$$J_i = \sum_{k=1}^{N_B} \left({}^{\mathrm{t}} \mathbf{f}_1^i \, \mathbf{b}_1^k \right) \cdot \left({}^{\mathrm{t}} \mathbf{f}_2^i \, \mathbf{b}_2^k \right) \cdot \ldots \cdot \left({}^{\mathrm{t}} \mathbf{f}_d^i \, \mathbf{b}_d^k \right).$$

The enrichment stage includes new candidates for enriching the reduced separated approximation basis, so that ψ can be updated as

$$\boldsymbol{\Psi} = \underbrace{\sum_{j=1}^{N_F} \alpha_j \, \mathbf{f}_1^j \otimes \mathbf{f}_2^j \otimes \cdots \otimes \mathbf{f}_d^j}_{=:\boldsymbol{\Psi}_{\mathbf{f}}} + \underbrace{\mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \cdots \otimes \mathbf{r}_d}_{=:\boldsymbol{\Psi}_{\mathbf{r}}}, \quad (A.1.6)$$

and (A.1.2) is replaced by the system

$${}^{\mathrm{t}}\boldsymbol{\psi}^{*}\boldsymbol{\mathcal{A}}\boldsymbol{\psi}_{\mathbf{r}} + {}^{\mathrm{t}}\boldsymbol{\psi}^{*}\boldsymbol{\mathcal{A}}\boldsymbol{\psi}_{\mathbf{f}} = {}^{\mathrm{t}}\boldsymbol{\psi}^{*}\boldsymbol{\mathcal{B}}.$$
 (A.1.7)

In particular, within a fixed point alternating direction algorithm, at each iteration we look for the computation of a single discrete function, \mathbf{r}_j , all the other components of $\psi_{\mathbf{r}}$ being assumed known, and after setting ψ^* to $\mathbf{r}_1 \otimes \cdots \mathbf{r}_{j-1} \otimes \mathbf{r}_j^* \otimes \mathbf{r}_{j+1} \cdots \otimes \mathbf{r}_d$. This strategy leads to solve the linear system

$$\mathbf{K}\,\mathbf{r}_j + \mathbf{v} = \tilde{\mathbf{v}},\tag{A.1.8}$$

where

$$\mathbf{K} = \sum_{k=1}^{N_A} \left(\mathbf{A}_j^k \prod_{\substack{h=1\\h\neq j}}^d {}^{\mathrm{t}} \mathbf{r}_h \, \mathbf{A}_h^k \, \mathbf{r}_h \right),$$
$$\mathbf{v} = \sum_{i=1}^{N_F} \sum_{k=1}^{N_A} \left(\alpha_i \, \mathbf{A}_j^k \, \mathbf{f}_j^i \prod_{\substack{h=1\\h\neq j}}^d {}^{\mathrm{t}} \mathbf{r}_h \, \mathbf{A}_h^k \, \mathbf{f}_h^i \right), \quad \tilde{\mathbf{v}} = \sum_{k=1}^{N_B} \left(\mathbf{b}_j^k \prod_{\substack{h=1\\h\neq j}}^d {}^{\mathrm{t}} \mathbf{r}_h \, \mathbf{b}_h^k \right).$$

For the explicit computations leading to system (A.1.8), we refer the interested reader to [31], whereas a convergence analysis for this greedy rank-1 update algorithm can be found in [232].

Remark A.1.1. The projection in (A.1.5) does not represent the only possible choice when defining $\boldsymbol{\psi}$. Alternatively, one could use a standard greedy-based enrichment [32,232]. The aim of the projection is to regularize the solution, thus getting rid of possible spurious modes yielded by an enrichment procedure and, consequently, to enhance the global convergence. Indeed, since modes are normalized, coefficients α_j in (A.1.5) can be intended as a truncation error since they express the importance of the different enrichment terms. This allows us to stop the enrichment procedure when not significant modes occur, thus limiting the inclusion of noisy information.

Remark A.1.2. As discussed in [31], the convergence of the fixed point strategy characterizing the enrichment step is guaranteed for symmetric discrete operators. Actually, numerical tests performed with non symmetric discrete operators (e.g., with hyperbolic operators) exhibit some difficulties in the convergence of the fixed point procedure as well as a loss of optimality in terms of number of function products to be computed. For this reason, all the problems analyzed in this work are suitably rewritten in order to deal with a symmetric operator. In general, the new problem to be solved reads:

$${}^{\mathrm{t}}\psi^{*}\,{}^{\mathrm{t}}\!\mathcal{A}\mathcal{A}\,\psi={}^{\mathrm{t}}\psi^{*}\,{}^{\mathrm{t}}\!\mathcal{A}\mathcal{B},$$

and the associated projection stage becomes

$$\sum_{j=1}^{N_F} H_{ij}^S \alpha_j = J_i^S \quad \forall i = 1, \dots, N_F$$

where

$$H_{ij}^{S} = \sum_{k'=1}^{N_{A}} \sum_{k=1}^{N_{A}} \left({}^{\mathrm{t}} \mathbf{f}_{1}^{i} {}^{\mathrm{t}} \mathbf{A}_{1}^{k'} \mathbf{A}_{1}^{k} \mathbf{f}_{1}^{j} \right) \cdot \left({}^{\mathrm{t}} \mathbf{f}_{2}^{i} {}^{\mathrm{t}} \mathbf{A}_{2}^{k'} \mathbf{A}_{2}^{k} \mathbf{f}_{2}^{j} \right) \cdot \ldots \cdot \left({}^{\mathrm{t}} \mathbf{f}_{d}^{i} {}^{\mathrm{t}} \mathbf{A}_{d}^{k'} \mathbf{A}_{d}^{k} \mathbf{f}_{d}^{j} \right),$$
$$J_{i}^{S} = \sum_{k'=1}^{N_{A}} \sum_{k=1}^{N_{B}} \left({}^{\mathrm{t}} \mathbf{f}_{1}^{i} {}^{\mathrm{t}} \mathbf{A}_{1}^{k'} \mathbf{b}_{1}^{k} \right) \cdot \left({}^{\mathrm{t}} \mathbf{f}_{2}^{i} {}^{\mathrm{t}} \mathbf{A}_{2}^{k'} \mathbf{b}_{2}^{k} \right) \cdot \ldots \cdot \left({}^{\mathrm{t}} \mathbf{f}_{d}^{i} {}^{\mathrm{t}} \mathbf{A}_{d}^{k'} \mathbf{b}_{d}^{k} \right).$$

Analogously, system (A.1.7) will be replaced by the new one

$$\mathbf{\psi}^{*} \, {}^{\mathrm{t}}\mathcal{A} \, \mathcal{A} \, \mathbf{\psi}_{\mathbf{r}} + {}^{\mathrm{t}} \mathbf{\psi}^{*} \, {}^{\mathrm{t}}\mathcal{A} \, \mathcal{A} \, \mathbf{\psi}_{\mathbf{f}} = {}^{\mathrm{t}} \mathbf{\psi}^{*} \, {}^{\mathrm{t}}\mathcal{A} \, \mathcal{B}_{\mathbf{f}}$$

so that system (A.1.8) takes the form

$$\mathbf{K}^S \, \mathbf{r}_j + \mathbf{v}^S = \tilde{\mathbf{v}}^S,$$

with

$$\begin{split} \mathbf{K}^{S} &= \sum_{k'=1}^{N_{A}} \sum_{k=1}^{N_{A}} \left({}^{\mathrm{t}} \mathbf{A}_{j}^{k'} \mathbf{A}_{j}^{k} \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}} \mathbf{r}_{h} {}^{\mathrm{t}} \mathbf{A}_{h}^{k'} \mathbf{A}_{h}^{k} \mathbf{r}_{h} \right), \\ \mathbf{v}^{S} &= \sum_{i=1}^{N_{F}} \sum_{k'=1}^{N_{A}} \sum_{k=1}^{N_{A}} \left(\alpha_{i} {}^{\mathrm{t}} \mathbf{A}_{j}^{k'} \mathbf{A}_{j}^{k} \mathbf{f}_{j}^{i} \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}} \mathbf{r}_{h} {}^{\mathrm{t}} \mathbf{A}_{h}^{k'} \mathbf{A}_{h}^{k} \mathbf{f}_{h}^{i} \right), \\ \tilde{\mathbf{v}}^{S} &= \sum_{k'=1}^{N_{A}} \sum_{k=1}^{N_{B}} \left({}^{\mathrm{t}} \mathbf{A}_{j}^{k'} \mathbf{b}_{j}^{k} \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}} \mathbf{r}_{h} {}^{\mathrm{t}} \mathbf{A}_{h}^{k'} \mathbf{b}_{h}^{k} \right). \end{split}$$

A.2 POD AND GAPPY-POD

A.2.1 POD

This section shortly revisits the main concept of the proper orthogonal decomposition (POD) [27] before introducing its gappy counterpart in section A.2.2.

Let us consider a full snapshot of $\psi(\boldsymbol{x},t)$ defined as

$$\Psi = (\Psi_1 | \cdots | \Psi_{N_t}) \in \mathbb{R}^{N_x \times N_t}$$
(A.2.1)

-

where ψ_j is a column vector containing the numerical approximations of $\psi(\boldsymbol{x}, t_j)$ in all the $N_{\boldsymbol{x}}$ spatial mesh points, for $j = 1, \ldots, N_t$.

The POD looks for the low-rank separated approximation

$$\Psi \approx \mathbf{W} \mathbf{\Lambda} \tag{A.2.2}$$

where

$$\mathbf{W} = \begin{pmatrix} \mathbf{w}_1 | & \cdots & | \mathbf{w}_m \end{pmatrix} \in \mathbb{R}^{N_x \times m}, \quad \mathbf{\Lambda} = \begin{pmatrix} \lambda_1^{\mathsf{I}} \\ \vdots \\ \lambda_m^{\mathsf{T}} \end{pmatrix} \in \mathbb{R}^{m \times N_t}.$$
(A.2.3)

Such decomposition is obtained computing the eigenvectors associated to the *m* highest eigenvalues of the correlation matrix $\mathbf{C} = \boldsymbol{\Psi}^{\mathsf{T}} \boldsymbol{\Psi} \in \mathbb{R}^{N_t \times N_t}$, accounting for the orthonormality condition $\boldsymbol{\Lambda}^{\mathsf{T}} \boldsymbol{\Lambda} = \mathbf{I}$. Once the temporal basis $\boldsymbol{\Lambda}$ is available, the spatial one is obtained by projection of the snapshots, as $\mathbf{W} = \boldsymbol{\Psi} \boldsymbol{\Lambda}^{\mathsf{T}}$.

Remark A.1. If the correlation matrix is defined as $\mathbf{C}^{\mathsf{T}} \in \mathbb{R}^{N_x \times N_x}$, its eigenvectors constitute the spatial orthonormal basis \mathbf{W} . Afterwards, the temporal one is obtained by projection in the same manner.

In a continuous framework, this procedure furnishes the approximation

$$\psi(\boldsymbol{x},t) \approx \psi^{\text{POD}}(\boldsymbol{x},t) = \sum_{k=1}^{m} w_k(\boldsymbol{x})\lambda_k(t),$$
 (A.2.4)

where the functions $w_k(\mathbf{x})$ and $\lambda_k(t)$ are the space and time modes expressed through the corresponding bases **W** and **A**.

A.2.2 GAPPY-POD

The gappy-POD [181, 182] follows the same strategy described in section A.2.1, but instead of extracting the reduced basis from the full snapshot (A.2.1), it considers only a gappy (incomplete) snapshot:

$$\bar{\boldsymbol{\Psi}} = \begin{pmatrix} \bar{\boldsymbol{\Psi}}_1 | & \cdots & | \bar{\boldsymbol{\Psi}}_{\bar{N}_t} \end{pmatrix} \in \mathbb{R}^{N_x \times \bar{N}_t}, \tag{A.2.5}$$

where the quantities denoted with $\overline{\bullet}$ are related to a subset of equispaced time instants $\{\overline{t}_1, \ldots, \overline{t}_{N_t}\} \subset \{t_1, \ldots, t_{N_t}\}.$

This starts by considering the (gappy) correlation matrix $\bar{\mathbf{C}}_g = \bar{\mathbf{\Psi}}^{\mathsf{T}} \bar{\mathbf{\Psi}} \in \mathbb{R}^{\bar{N}_t \times \bar{N}_t}$ and by solving the eigenvalue problem to compute the (gappy) temporal orthonormal basis $\bar{\mathbf{\Lambda}}_g \in \mathbb{R}^{m \times \bar{N}_t}$. Then, the spatial functions are computed as $\bar{\mathbf{W}} = \bar{\mathbf{\Psi}} \bar{\mathbf{\Lambda}}_g^{\mathsf{T}} \in \mathbb{R}^{N_x \times \bar{m}}$. It shall be noticed that the basis $\overline{\mathbf{W}}$ is defined over all the degrees of freedom, and for this reason we drop the subscript g, which tracks the quantities defined on incomplete sets.

Once $\overline{\mathbf{W}}$ is available, an approximation of the temporal functions $\overline{\mathbf{\Lambda}}$ is recovered solving

$$\min_{\bar{\mathbf{\Lambda}}\in\mathbb{R}^{\bar{m}\times N_t}} \left\|\bar{\mathbf{W}}\bar{\mathbf{\Lambda}}-\mathbf{\Psi}\right\|_2,\tag{A.2.6}$$

whose least-square solution reads $\bar{\mathbf{\Lambda}} = (\bar{\mathbf{W}}^{\mathsf{T}} \bar{\mathbf{W}})^{-1} \bar{\mathbf{W}}^{\mathsf{T}} \Psi \in \mathbb{R}^{m \times N_t}$. In this way the gappy data in time $\bar{\mathbf{\Lambda}}_g$ can been approximated over all N_t time steps.

Equivalently to (A.2.4), the Gappy-POD continuous approximation can be written as

$$\psi(\boldsymbol{x},t) \approx \psi^{\text{GPOD}}(\boldsymbol{x},t) = \sum_{k=1}^{\bar{m}} \bar{w}_k(\boldsymbol{x})\bar{\lambda}_k(t).$$
 (A.2.7)

Once the approximation (A.2.7) is computed, the time modes allow to study quickly the transient behaviour. This consists in determining a threshold T_i such that, for $t > T_i$ the functions $\psi(\mathbf{x}, t)$ clearly exhibit a multiscale behaviour.

At this point, a fully separated space-macrotime-microtime PGD approximation can be efficiently computed [54, 55], that is

$$\psi(\boldsymbol{x}, t) \mathbb{1}_{(T_i, T_K)}(t) \approx \sum_{k=1}^m \Psi_k^{\boldsymbol{x}}(\boldsymbol{x}) \Psi_k^{\boldsymbol{\tau}}(\tau) \Psi_k^{\boldsymbol{T}}(T).$$
(A.2.8)

where $\mathbb{1}_{I}(t)$ denotes the indicator function of the set I.

Remark A.2. It shall be noticed that often $\psi(\boldsymbol{x},t)$ is a vector-valued function, having D > 1 components $(\psi^{(d)}(\boldsymbol{x},t))_{d=1}^{D}$. In this case, one could assemble D by-component snapshots $\Psi^{(d)} \in \mathbb{R}^{N_x \times N_t}$ or a unique snapshot $\Psi \in \mathbb{R}^{DN_x \times N_t}$, where the column vectors account for the concatenation of those components. For instance, this happens when the quantity of interest is the plastic strain tensor, which, in a two-dimensional spatial domain, has the three components $\boldsymbol{\varepsilon}^p = (\varepsilon_{xx}^p, \varepsilon_{yy}^p, \varepsilon_{xy}^p)$.

A.3 HODMD

The DMD-d algorithm basically consists of applying the classical DMD(-1) to enlarged snapshots, following several steps [171, 179, 180]:

1. **Snapshots enlargement.** Equation (3.3.33) can be rewritten in terms of standard Koopman assumption

$$\tilde{\mathbf{v}}_{j+1} \approx \mathbf{R} \tilde{\mathbf{v}}_j,$$
 (A.3.1)

involving enlarged snapshots and (unknown) Koopman matrix

$$\tilde{\mathbf{v}}_{j} = \begin{pmatrix} v_{j} \\ v_{j+1} \\ \vdots \\ v_{j+d-2} \\ v_{j+d-1} \end{pmatrix} \in \mathbb{R}^{d}, \quad \tilde{\mathbf{R}} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 \\ c_{1} & c_{2} & c_{3} & \cdots & c_{d-1} & c_{d} \end{pmatrix} \in \mathbb{R}^{d \times d}, \quad (A.3.2)$$

with $1 \le j \le K - d$.

The algorithm considers the DMD(-1) to the modified snapshot matrix

$$\tilde{\mathbf{V}}_{1}^{K-d+1} = \left(\tilde{\mathbf{v}}_{1} | \cdots | \tilde{\mathbf{v}}_{K-d+1}\right) \in \mathbb{R}^{d \times (K-d+1)}, \quad (A.3.3)$$

where each column $\tilde{\mathbf{v}}_j$ is defined as in equation (A.3.2). The subscript and the superscript notation simply tracks the first and the last value of index j.

Actually this can be seen as superimposed DMD in a sliding window including d time-steps.

2. Snapshots reduction. As usual in the DMD, from the snapshot matrix (A.3.3), a truncated SVD of rank $r \leq \min\{d, K - d + 1\} = d$ is performed, giving

$$\tilde{\mathbf{V}}_1^{K-d+1} \approx \mathbf{U}_r \hat{\mathbf{V}}_1^{K-d+1} \tag{A.3.4}$$

where

$$\hat{\mathbf{V}}_{1}^{K-d+1} = \boldsymbol{\Sigma}_{r} \mathbf{T}_{r}^{\mathsf{T}} = \begin{pmatrix} \hat{\mathbf{v}}_{1} | & \cdots & | \hat{\mathbf{v}}_{K-d+1} \end{pmatrix} \in \mathbb{R}^{r \times (K-d+1)}$$
(A.3.5)

with $\Sigma_r \in \mathbb{R}^{r \times r}$, $\mathbf{U}_r \in \mathbb{R}^{d \times r}$, $\mathbf{T}_r \in \mathbb{R}^{(K-d+1) \times r}$ the usual matrices outcome of a reduced SVD. The relation between the enlarged snapshots and the enlargedreduced ones is simply given by the projection matrix as

$$\tilde{\mathbf{v}}_j = \mathbf{U}_r \hat{\mathbf{v}}_j, \quad \hat{\mathbf{v}}_j = \mathbf{U}_r^{\mathsf{T}} \tilde{\mathbf{v}}_j.$$
 (A.3.6)

At this point, pre-multiplying the Koopman assumption equation (A.3.1) by $\mathbf{U}_r^{\mathsf{T}}$, since $\mathbf{U}_r^{\mathsf{T}}\mathbf{U}_r = \mathbf{I}_r$ by orthogonality, we obtain

$$\hat{\mathbf{v}}_{j+1} \approx \hat{\mathbf{R}} \hat{\mathbf{v}}_j,$$
 (A.3.7)

where $\hat{\mathbf{R}} = \mathbf{U}_r^{\mathsf{T}} \tilde{\mathbf{R}} \mathbf{U}_r$ is the enlarged-reduced snapshot matrix.

Remark A.3. It shall be noticed that in [179, 180] two reduction steps were performed since the procedure was applied to a vector-valued time series, while we are only interested in sequences of scalars.

3. **DMD modes computation.** This step consists in determining an approximation of the enlarged-reduced Koopman matrix $\hat{\mathbf{R}}$. Relation (A.3.7) can be written in terms of enlarged-reduced matrix snapshots as

$$\hat{\mathbf{V}}_2^{K-d+1} \approx \hat{\mathbf{R}} \hat{\mathbf{V}}_1^{K-d} \tag{A.3.8}$$

At this point the HODMD expansion is obtained by applying the classic DMD algorithm to equation (A.3.8). This consists in applying the standard SVD to the matrix $\hat{\mathbf{V}}_{1}^{K-d} \in \mathbb{R}^{r \times (K-d)}$:

$$\hat{\mathbf{V}}_{1}^{K-d} = \hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{T}}^{\mathsf{T}} \tag{A.3.9}$$

with $\hat{\boldsymbol{\Sigma}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{U}} \in \mathbb{R}^{r \times r}$, $\hat{\mathbf{T}} \in \mathbb{R}^{(K-d+1) \times r}$ the usual matrices outcome of a standard SVD, verifying $\hat{\mathbf{U}}^{\mathsf{T}}\hat{\mathbf{U}} = \hat{\mathbf{U}}\hat{\mathbf{U}}^{\mathsf{T}} = \mathbf{I}_r$, $\hat{\mathbf{T}}^{\mathsf{T}}\hat{\mathbf{T}} = \mathbf{I}_r$ and $\hat{\boldsymbol{\Sigma}}$ is non-singular. Now, (A.3.9) is inserted in equation (A.3.8) to obtain

$$\hat{\mathbf{V}}_{2}^{K-d+1} \approx \hat{\mathbf{R}}\hat{\mathbf{U}}\hat{\boldsymbol{\Sigma}}\hat{\mathbf{T}}^{\mathsf{T}},\tag{A.3.10}$$

which is simply post-multiplied by $\hat{\mathbf{T}}\hat{\boldsymbol{\Sigma}}^{-1}\hat{\mathbf{U}}^{\mathsf{T}}$ to obtain an approximation of the enlarged-reduced Koopman matrix

$$\hat{\mathbf{R}} \approx \hat{\mathbf{V}}_2^{K-d+1} \hat{\mathbf{T}} \hat{\mathbf{\Sigma}}^{-1} \hat{\mathbf{U}}^{\mathsf{T}}.$$
(A.3.11)

Once $\hat{\mathbf{R}}$ is approximated, equation (A.3.7) is solved via the expansion

$$\hat{\mathbf{v}}_j \approx \hat{\mathbf{v}}_j^{\text{DMD}} = \sum_{m=1}^r a_m \hat{\mathbf{q}}_m \mu_m^{j-1}$$
(A.3.12)

with j = 1, ..., K. In this expression $(\hat{\mathbf{q}}_m, \mu_m)$ are the couples eigenvectoreigenvalue of $\hat{\mathbf{R}}$.

Moreover, (A.3.12) may be rewritten as an exponential expansion in terms of frequencies ω_m and growth rates δ_m

$$\hat{\mathbf{v}}_j \approx \hat{\mathbf{v}}_j^{\text{DMD}} = \sum_{m=1}^r a_m \hat{\mathbf{q}}_m e^{(\delta_m + \omega_m i)(j-1)\Delta T}$$
(A.3.13)

where

$$\delta_m + \omega_m i = \frac{1}{\Delta T} \log \mu_m \tag{A.3.14}$$

Of course, the DMD expansion for the enlarged snapshots is obtained premultiplying (A.3.13) by \mathbf{U}_r :

$$\tilde{\mathbf{v}}_j \approx \tilde{\mathbf{v}}_j^{\text{DMD}} = \sum_{m=1}^r a_m \tilde{\mathbf{q}}_m e^{(\delta_m + \omega_m i)(j-1)\Delta T}$$
(A.3.15)

where $\tilde{\mathbf{q}}_m = \mathbf{U}_r \hat{\mathbf{q}}_m$.

4. **DMD amplitudes computation.** The computation of the amplitudes a_m is done in the usual manner of the DMD procedure and, for the sake of brevity, we refer to [179], whose notation was followed in the whole section.

IMPLEMENTATION ASPECTS OF THE NURBS-PGD METHOD

B.1 NURBS-BASED GEOMETRY DESCRIPTION

A local bijection is constructed through NURBS as follows

$$\boldsymbol{F}: \Omega_{\boldsymbol{\xi}} \to \Omega_{\boldsymbol{x}}$$
$$\boldsymbol{\xi} \mapsto \boldsymbol{x} = \boldsymbol{F}(\boldsymbol{\xi}) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{ij}^{pq}(\boldsymbol{\xi}) \boldsymbol{B}_{ij},$$
(B.1.1)

which represents a one to one relation that maps any point $\boldsymbol{\xi}$ in the computational domain $\Omega_{\boldsymbol{\xi}}$ to a point \boldsymbol{x} in physical domain $\Omega_{\boldsymbol{x}}$.

In Eq. (B.1.1), **B** refers to the control points (the vertices of the so-called control net) in the physical domain and $R_{ij}^{pq}(\xi,\eta)$ denote the bivariate NURBS basis functions in 2D domains. Such functions can be obtained using tensor product of univariate basis functions, as follows

$$R_{ij}^{pq}(\boldsymbol{\xi}) = \frac{N_{ip}(\xi)N_{jq}(\eta)w_{ij}}{\sum_{\alpha=1}^{n}\sum_{\beta=1}^{m}N_{\alpha p}(\xi)N_{\beta q}(\eta)w_{\alpha\beta}}.$$
(B.1.2)

In the above equation w_{ij} denotes the geometry related weight of the combination and subscripts p and q denote the order of B-splines in directions ξ and η , respectively. $N_{ap}(\xi)$ is the univariate B-spline basis function of order p in the a-th knot span $\xi \in [\xi_a, \xi_{a+1}]$. The following recursive equations can be used to compute the univariate B-spline basis function $N_{ap}(\xi)$ [234]

$$N_{a0}(\xi) = \begin{cases} 1 & \xi_a \le \xi < \xi_{a+1} \\ 0 & \text{otherwise} \end{cases}, \quad \text{for } p = 0 \tag{B.1.3}$$

$$N_{ap}(\xi) = \frac{\xi - \xi_a}{\xi_{a+p} - \xi_a} N_{a(p-1)} + \frac{\xi_{a+p+1} - \xi}{\xi_{a+p+1} - \xi_{a+1}} N_{(a+1)(p-1)} , \quad \text{for } p > 0.$$
(B.1.4)

To define a set of *n* B-spline functions of order *p* in a univariate parametric space $\xi \in [0, 1]$, the knot vector κ_{ξ} is defined as follows.

$$\boldsymbol{\kappa}_{\xi} = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]^T.$$
 (B.1.5)

Same holds for the knot vector κ_{η} applying in the coordinates η read

$$\boldsymbol{\kappa}_{\eta} = [\eta_1, \eta_2, \dots, \eta_{m+q+1}]^T, \tag{B.1.6}$$

being m the number of basis functions of order q in the direction η .

A proper choice of the knot vectors allows obtaining rich behavior of the basis functions and enough flexibility to describe complex geometries.

Now, using the notation $(x, y) \equiv (x_1, x_2)$ and $(\xi, \eta) \equiv (\xi_1, \xi_2)$, for facilitating the mapping description compactness, the terms involved in the transformation of the differential operator from Ω_x to Ω_{ξ} read

$$\frac{\partial \boldsymbol{x}}{\partial \xi_a} = \sum_{i=1}^n \sum_{j=1}^m \frac{\partial R_{ij}^{pq}(\boldsymbol{\xi})}{\partial \xi_a} \boldsymbol{B}_{ij}, \qquad a, b = 1, 2.$$
(B.1.7)

B.2 SEPARATION OF SPACE: SEPARATED APPROXIMATE REPRESENTATION -SAR-

One of the most appealing features of the PGD is the reduction of a high dimensional problem to a series of problems defined in lower dimensional spaces [235, 236]. Since the PGD exploits separation of variables, to maximize the accuracy of the method it is preferable to work in separable domains.

The NURBS-based geometric mapping presented in subsection B.1 allows to map a non-regular geometry Ω_x into the separable computational domain Ω_{ξ} , via mapping (B.1.1). However, one needs also to map the differential problem \mathcal{P}_x from the physical domain Ω_x accounting for the Jacobian of the transformation B.1.1, that is (B.1.7):

$$\begin{bmatrix} \partial(\cdot)/\partial x\\ \partial(\cdot)/\partial y \end{bmatrix} = h \begin{bmatrix} \partial(\cdot)/\partial \xi\\ \partial(\cdot)/\partial \eta \end{bmatrix},$$
 (B.2.1)

where h is the inverse of the Jacobian tensor, J.

Once the mapping (4.2.2) is introduced, $\mathcal{P}_{\boldsymbol{x}}$ can be mapped back to the fixed domain $\overline{\Omega} \equiv \Omega_{\boldsymbol{\xi}}$, leading to problem $\mathcal{P}_{\boldsymbol{\xi}}$.

Even if the computational domain $\Omega_{\boldsymbol{\xi}}$ is fully separable by construction, the differential operators are not, because of the fact that shape functions involved in the geometry mapping are not separated. Indeed, the geometric mapping given in Eq. (B.1.1) and its derivatives in Eq. (B.1.7) leads to a non-separable mapping in the sense of the PGD technique, compromising the effectiveness of the PGD solver. To overcome this difficulty, a separated approximate representation (SAR) of the Jacobian of the transformation [58].

To briefly illustrate the SAR, let us consider a generic function $g(\boldsymbol{x})$ defined in the physical domain $\Omega_{\boldsymbol{x}}$ and its counterpart, $g(\boldsymbol{\xi})$, in the computational domain $\Omega_{\boldsymbol{\xi}}$. A separated approximate representation of $g(\boldsymbol{\xi})$ in the computational domain $\Omega_{\boldsymbol{\xi}}$ reads

$$g(\boldsymbol{\xi}) \approx \sum_{i=1}^{N_G} \prod_{j=1}^{N_D} \boldsymbol{M}_j^T(\xi_j) \boldsymbol{G}_{ji}, \qquad (B.2.2)$$

where, $M_j(\xi_j)$ is the vector of approximation functions in term of *j*-th coordinate direction and G_{ji} is the vector of coefficients of the *i*-th mode in the *j*-th coordinate direction. N_D is the total number of problem dimensions to be separated (i.e., 2 in this example) and N_G is the number of modes which are used to represent the separated representation of function $g(\boldsymbol{\xi})$. Generally, any type of approximation functions $M_j(\xi_j)$ could be used to approximate the modes in each direction but the piecewise linear Lagrange interpolation functions (which are common in the FEM) are used here for simplicity. More details on the procedure for calculating the unknown coefficients vector G_{ji} can be found in [58, 229, 235].

Thus, the SAR proceeds by enforcing in a weak form the equality $g(\boldsymbol{\xi}) = g(\boldsymbol{x}(\boldsymbol{\xi}))$ in $\Omega_{\boldsymbol{\xi}}$, with $g(\boldsymbol{\xi})$ expressed in the separated form (B.2.2), following the procedure deeply addressed in our former works [58, 235], to finally obtain the coefficients \boldsymbol{G}_{ji} .

The construction of a separated representation of the coordinates mapping $x(\xi)$ is obtained as follows:

$$\begin{cases} x(\boldsymbol{\xi}) = \sum_{i=1}^{N_X} \prod_{j=1}^{N_D} \boldsymbol{M}_j^T \boldsymbol{X}_{ji}, \\ y(\boldsymbol{\xi}) = \sum_{i=1}^{N_Y} \prod_{j=1}^{N_D} \boldsymbol{M}_j^T \boldsymbol{Y}_{ji}. \end{cases}$$
(B.2.3)

where, N_X and N_Y are the number of modes required to reach the desired level of accuracy in the SAR of $x(\boldsymbol{\xi})$ and $y(\boldsymbol{\xi})$, respectively. Using the same rationale, we can derive the SAR of Jacobian determinant, $|\boldsymbol{J}(\boldsymbol{\xi})|$, and the all 4 elements of the transformation derivatives tensor, $\boldsymbol{h}(\boldsymbol{\xi})$, as follows:

$$|\boldsymbol{J}| = \sum_{i=1}^{N_J} \prod_{j=1}^{N_D} \boldsymbol{M}_j^T \boldsymbol{J}_{ji}, \qquad (B.2.4)$$

and

$$h_{ab}(\boldsymbol{\xi}) = \sum_{i=1}^{N_{H}ab} \prod_{j=1}^{N_D} \boldsymbol{M}_j^T \boldsymbol{H}_{ji}^{ab}, \quad a, b = 1, 2.$$
(B.2.5)

More details regarding separated geometry mapping and its performance in parametric solution of field problems are given in [58].

BIBLIOGRAPHY

- [1] M. RAPPAZ, M. BELLET, AND M. DEVILLE, Numerical Modeling in Materials Science and Engineering, Springer series in computational mathematics; volume 32, Springer Verlag, 2003. Translated from the French: "Modélisation numérique en science et génie des matériaux". Contents: Preface.- Continuous Media.- The Finite Different Method.- Elements of Numerical Algorithms.-Phase Transformations.- Deformation of Solids.- Incompressible Fluid Flow.-Inverse Methods.- Stochastic Methods.- Appendices.- Index.
- [2] D. GIANNI, A. D'AMBROGIO, AND A. TOLK, eds., Modeling and Simulation-Based Systems Engineering Handbook, CRC Press, 2014.
- [3] M. S. SHEPHARD, M. W. BEALL, R. M. O'BARA, AND B. E. WEBSTER, *Toward simulation-based design*, Finite Elements in Analysis and Design, 40 (2004), pp. 1575–1598. The Fifteenth Annual Robert J. Melosh Competition.
- [4] J. D. HOFFMAN AND S. FRANKEL, Numerical methods for engineers and scientists, CRC press, 2018.
- [5] R. HAMMING, Numerical methods for scientists and engineers, Courier Corporation, 2012.
- [6] Volume 1 System- and Data-Driven Methods and Algorithms, De Gruyter, Berlin, Boston, 2021.
- [7] Volume 2 Snapshot-Based Methods and Algorithms, De Gruyter, Berlin, Boston, 2021.
- [8] W. PIETRASZKIEWICZ AND C. SZYMCZAK, Shell Structures, Theory and Applications: Proceedings of the 8th International Conference on Shell Structures (SSTA 2005), 12-14 October 2005, Jurata, Gdansk, Poland, no. vol. 2 in Balkema proceedings and monographs in engineering, water and earth sciences, Taylor & Francis, 2005.
- [9] E. OÑATE, Structural analysis with the finite element method. Linear statics. Volume 2: Beams, plates and shells, 01 2010.
- [10] M. FARSHAD, Design and Analysis of Shell Structures, Solid Mechanics and Its Applications, Springer Netherlands, 2013.
- [11] Y.-C. FUNG, P. TONG, AND X. CHEN, Classical and computational solid mechanics, World Scientific, 2001.
- [12] A. SHARMA, M. C. OH, AND B. AHN, Recent advances in very high cycle fatigue behavior of metals and alloys—a review, Metals, 10 (2020).
- [13] C.-H. HSUEH, S. SCHMAUDER, C.-S. CHEN, K. CHAWLA, W. CHEN, AND Y. KAGAWA, Handbook of Mechanics of Materials, 01 2019.
- [14] J. SHIJVE, Fatigue of structures and materials, Delft: Springer, (2009), p. 621.
- [15] J. FISH AND Q. YU, Computational mechanics of fatigue and life predictions for composite materials and structures, Computer methods in applied mechanics and engineering, 191 (2002), pp. 4827–4849.
- [16] R. BRANCO, F. BERTO, AND S. WU, Computational methods for fatigue and fracture, Metals, 12 (2022).
- [17] L. C. EVANS, Partial differential equations, vol. 19, American Mathematical Society, 2022.
- [18] S. SALSA, Partial Differential Equations in Action: From Modelling to Theory - UNITEX), Springer Publishing Company, Incorporated, 2nd ed., 2015.
- [19] L. DEBNATH AND L. DEBNATH, Nonlinear partial differential equations for scientists and engineers, Springer, 2005.
- [20] A. QUARTERONI AND A. VALLI, Numerical approximation of partial differential equations, vol. 23, Springer Science & Business Media, 2008.
- [21] J. W. THOMAS, Numerical partial differential equations: finite difference methods, vol. 22, Springer Science & Business Media, 2013.
- [22] T. J. HUGHES, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.
- [23] S. MAZUMDER, Numerical methods for partial differential equations: finite difference and finite volume methods, Academic Press, 2015.
- [24] S. A. SAUTER, C. SCHWAB, S. A. SAUTER, AND C. SCHWAB, Boundary element methods, Springer, 2011.
- [25] J. SHEN, T. TANG, AND L.-L. WANG, Spectral methods: algorithms, analysis and applications, vol. 41, Springer Science & Business Media, 2011.
- [26] J. A. COTTRELL, T. J. HUGHES, AND Y. BAZILEVS, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.
- [27] L. SIROVICH, Turbulence and the dynamics of coherent structures. i coherent structures. ii - symmetries and transformations. iii - dynamics and scaling, Quarterly of Applied Mathematics - QUART APPL MATH, 45 (1987).
- [28] G. BERKOOZ, P. HOLMES, AND J. L. LUMLEY, The proper orthogonal decomposition in the analysis of turbulent flows, Annual Review of Fluid Mechanics, 25 (1993), pp. 539–575.
- [29] F. CHINESTA, R. KEUNINGS, AND A. LEYGUE, The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer, 01 2014.
- [30] A. NOUY, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1603–1626.

- [31] A. AMMAR, The proper generalized decomposition: a powerful tool for model reduction, International Journal of Material Forming, 3 (2010), pp. 89–102.
- [32] A. AMMAR, F. CHINESTA, P. DÍEZ, AND A. HUERTA, An error estimator for separated representations of highly multidimensional models, Computer Methods in Applied Mechanics and Engineering - COMPUT METHOD APPL MECH ENG, 199 (2010), pp. 1872–1880.
- [33] F. CHINESTA, A. LEYGUE, F. BORDEU, J. V. AGUADO, E. CUETO, D. GON-ZALEZ, I. ALFARO, A. AMMAR, AND A. HUERTA, *Pgd-based computational* vademecum for efficient design, optimization and control, Archives of Computational Methods in Engineering, 20 (2013).
- [34] F. CHINESTA, A. AMMAR, AND E. CUETO, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Archives of Computational Methods in Engineering, 17 (2010), pp. 327– 350.
- [35] P. DÍEZ, S. ZLOTNIK, A. GARCÍA, AND A. HUERTA, Encapsulated pgd algebraic toolbox operating with high-dimensional data, Archives of Computational Methods in Engineering, 27 (2019).
- [36] B. BOGNET, F. BORDEU, F. CHINESTA, A. LEYGUE, AND A. POITOU, Advanced simulation of models defined in plate geometries : 3d solutions with 2d computational complexity, Computer Methods in Applied Mechanics and Engineering, s 201-204 (2012), p. 1-12.
- [37] B. BOGNET, A. LEYGUE, AND F. CHINESTA, Separated representations of 3d elastic solutions in shell geometries, Advanced Modeling and Simulation in Engineering Sciences, 1 (2014), pp. 1–34.
- [38] C. GHNATIOS, B. BOGNET, A. BARASINSKI, AND F. CHINESTA, Proper Generalized Decomposition Based Model Order Reduction Applied to Problems Defined in Degenerated Domains, 06 2021, pp. 1–29.
- [39] P. VIDAL, G. LAURENT, AND O. POLIT, Comparison of different degenerated approaches for the modeling of composite shell structures, Finite Elements in Analysis and Design, 195 (2021), p. 103585.
- [40] G. QUARANTA, M. ZIANE, E. HAUG, J.-L. DUVAL, AND F. CHINESTA, A minimally-intrusive fully 3d separated plate formulation in computational structural mechanics, Advanced Modeling and Simulation in Engineering Sciences, 6 (2019), p. 11.
- [41] G. QUARANTA, M. ZIANE, F. DAIM, E. ABISSET-CHAVANNE, J.-L. DUVAL, AND F. CHINESTA, On the coupling of local 3d solutions and global 2d shell theory in structural mechanics, Advanced Modeling and Simulation in Engineering Sciences, 6 (2019).
- [42] G. QUARANTA, B. BOGNET, R. IBAÑEZ, A. TRAMECON, E. HAUG, AND F. CHINESTA, A new hybrid explicit/implicit in-plane-out-of-plane separated representation for the solution of dynamic problems defined in plate-like domains, Computers & Structures, 210 (2018), pp. 135–144.

- [43] C. GHNATIOS, F. CHINESTA, AND C. BINETRUY, 3d modeling of squeeze flows occurring in composite laminates, International Journal of Material Forming, 8 (2015), pp. 73–83.
- [44] C. GHNATIOS, E. ABISSET-CHAVANNE, C. BINETRUY, F. CHINESTA, AND S. ADVANI, 3d modeling of squeeze flow of multiaxial laminates, Journal of Non-Newtonian Fluid Mechanics, 234 (2016).
- [45] C. GHNATIOS, E. ABISSET-CHAVANNE, C. BINETRUY, F. CHINESTA, AND S. ADVANI, 3d modeling of squeeze flow of unidirectionally thermoplastic composite inserts, vol. 1769, 10 2016, p. 170002.
- [46] A. AMMAR, C. GHNATIOS, F. DELPLACE, A. BARASINSKI, J.-L. DUVAL, E. CUETO, AND F. CHINESTA, On the effective conductivity and the apparent viscosity of a thin rough polymer interface using pgd-based separated representations, International Journal for Numerical Methods in Engineering, 121 (2020), pp. 5256–5274.
- [47] F. CHINESTA, A. AMMAR, A. LEYGUE, AND R. KEUNINGS, An overview of the proper generalized decomposition with applications in computational rheology, Journal of Non-Newtonian Fluid Mechanics, 166 (2011), p. 578–592.
- [48] D. CANALES, A. LEYGUE, F. CHINESTA, I. ALFARO, D. GONZÁLEZ, E. CUETO, ÉRIC FEULVARCH, AND J.-M. BERGHEAU, In-plane/out-of-plane separated representations of updated lagrangian descriptions of viscoplastic flow models in plate domains, Comptes Rendus Mécanique, 344 (2016), pp. 225–235. Computational simulation of manufacturing processes.
- [49] H. TERTRAIS, R. IBAÑEZ, A. BARASINSKI, C. GHNATIOS, AND F. CHINESTA, On the proper generalized decomposition applied to microwave processes involving multilayered components, Mathematics and Computers in Simulation, 156 (2019), pp. 347–363.
- [50] A. BARASINSKI, C. GHNATIOS, E. ABENIUS, S. BECHTEL, AND F. CHINESTA, Electromagnetic field propagation in a composite laminate and induced thermal field, International Journal of Material Forming, 14 (2021), pp. 97–103.
- [51] A. LEON, S. MUELLER, P. DE LUCA, R. SAID, J.-L. DUVAL, AND F. CHINESTA, Non-intrusive proper generalized decomposition involving space and parameters: application to the mechanical modeling of 3d woven fabrics, Advanced Modeling and Simulation in Engineering Sciences, 6 (2019), p. 13.
- [52] C. GERMOSO, G. QUARANTA, J. L. DUVAL, AND F. CHINESTA, Non-Intrusive In-Plane-Out-of-Plane Separated Representation in 3D Parametric Elastodynamics, Computation, 8 (2020), pp. 1–15.
- [53] A. SCHMID, A. PASQUALE, C. ELLERSDORFER, M. ZIANE, M. RAFFLER, V. CHAMPANEY, F. FEIST, AND F. CHINESTA, Application of pgd separation of space to create a reduced-order model of a lithium-ion cell structure, Frontiers in Materials, 10 (2023).
- [54] A. PASQUALE, A. AMMAR, A. FALCO, S. PEROTTO, E. CUETO, J.-L. DUVAL, AND F. CHINESTA, A separated representation involving multiple time scales within the proper generalized decomposition framework, Advanced Modeling and Simulation in Engineering Sciences, 8 (2021), p. 26.

- [55] F. CHINESTA, A. AMMAR, AND E. CUETO, Proper generalized decomposition of time-multiscale models, International Journal for Numerical Methods in Engineering, 83 (2010), pp. 1114 – 1132.
- [56] S. RODRIGUEZ ITURRA, Virtual charts for seismic engineering including parameters associated with loading, PhD thesis, 2021. Thèse de doctorat dirigée par Néron David, Ladevèze Pierre et Charbonnel Pierre-Etienne. Génie civil université Paris-Saclay 2021.
- [57] F. CHINESTA, P. LADEVÈZE, AND E. CUETO, A short review on model order reduction based on proper generalized decomposition, Archives of Computational Methods in Engineering, 18 (2011), pp. 395–404.
- [58] M.-J. KAZEMZADEH-PARSI, A. AMMAR, J. DUVAL, AND F. CHINESTA, Enhanced parametric shape descriptions in pgd-based space separated representations, Advanced Modeling and Simulation in Engineering Sciences, 8 (2021).
- [59] M.-J. KAZEMZADEH-PARSI, A. PASQUALE, D. DI LORENZO, V. CHAMPANEY, A. AMMAR, AND F. CHINESTA, Nurbs-based shape parametrization enabling pgd-based space separability: Methodology and application, Finite Elements in Analysis and Design, 227 (2023), p. 104022.
- [60] L. CHAMOIN AND H. THAI, Certified real-time shape optimization using isogeometric analysis, pgd model reduction, and a posteriori error estimation, International Journal for Numerical Methods in Engineering, 119 (2019), pp. 151–176.
- [61] A. SANCARLOS, V. CHAMPANEY, E. CUETO, AND F. CHINESTA, Regularized regressions for parametric models based on separated representations, Advanced Modeling and Simulation in Engineering Sciences, 10 (2023).
- [62] V. CHAMPANEY, A. PASQUALE, A. AMMAR, AND F. CHINESTA, Parametric curves metamodelling based on data clustering, data alignment, pod-based modes extraction and pgd-based nonlinear regressions, Frontiers in Materials, 9 (2022), p. 904707.
- [63] A. PASQUALE, V. CHAMPANEY, Y. KIM, N. HASCOËT, A. AMMAR, AND F. CHINESTA, A parametric metamodel of the vehicle frontal structure accounting for material properties and strain-rate effect: application to full frontal rigid barrier crash test, Heliyon, 8 (2022), p. e12397.
- [64] A. SCHMID, A. PASQUALE, C. ELLERSDORFER, V. CHAMPANEY, M. RAF-FLER, S. GUÉVELOU, S. KIZIO, M. ZIANE, F. FEIST, AND F. CHINESTA, Pgd based meta modelling of a lithium-ion battery for real time prediction, Frontiers in Materials (Computational Materials Science), 10 (2023).
- [65] V. CHAMPANEY, A. SANCARLOS, F. CHINESTA, E. CUETO, D. GONZÁLEZ, I. ALFARO, S. GUÉVELOU, J. DUVAL, A. CHAMBARD, AND P. MOURGUE, Hybrid twins - a highway towards a performance-based engineering. part i: Advanced model order reduction enabling real-time physics, ESAFORM 2021, (2021).
- [66] F. CHINESTA, E. CUETO, AND S. GUEVELOU, Material forming digital twins: The alliance between physics-based and data-driven models, Key Engineering Materials, 926 (2022), pp. 3–14.

- [67] V. CHAMPANEY, F. CHINESTA, AND E. CUETO, Engineering empowered by physics-based and data-driven hybrid models: A methodological overview, International Journal of Material Forming, 15 (2022).
- [68] M. EL FALLAKI IDRISSI, F. PRAUD, V. CHAMPANEY, F. CHINESTA, AND F. MERAGHNI, Multiparametric modeling of composite materials based on non-intrusive pgd informed by multiscale analyses: Application for real-time stiffness prediction of woven composites, Composite Structures, 302 (2022), p. 116228.
- [69] V. LIMOUSIN, X. DELGERIE, E. LEROY, R. IBÁÑEZ, C. ARGERICH, F. DAIM, J. LOUIS DUVAL, AND F. CHINESTA, Advanced model order reduction and artificial intelligence techniques empowering advanced structural mechanics simulations: application to crash test analyses, Mechanics and Industry, 20 (2020), p. 804.
- [70] M. N. SHAHUL HAMEED, F. BORDEU, A. LEYGUE, AND F. CHINESTA, Arlequin based pgd domain decomposition, Computational Mechanics, 54 (2014).
- [71] A. HUERTA, E. NADAL SORIANO, AND F. CHINESTA, Proper generalized decomposition solutions within a domain decomposition strategy: Pgd solutions within a dd strategy, International Journal for Numerical Methods in Engineering, 113 (2017).
- [72] M. DISCACCIATI, B. J. EVANS, AND M. GIACOMINI, An overlapping domain decomposition method for the solution of parametric elliptic problems via proper generalized decomposition, Computer Methods in Applied Mechanics and Engineering, 418 (2024), p. 116484.
- [73] J. B. MACQUEEN, Some methods for classification and analysis of multivariate observations, in Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, L. M. L. Cam and J. Neyman, eds., vol. 1, University of California Press, 1967, pp. 281–297.
- [74] A. SCHMID, A. PASQUALE, C. ELLERSDORFER, M. RAFFLER, V. CHAM-PANEY, M. ZIANE, F. CHINESTA, AND F. FEIST, Mechanical characterization of li-ion cells and the calibration of numerical models using proper generalized decomposition, Proceedings of the ASME 2023 International Mechanical Engineering Congress and Exposition IMECE2023, (2023).
- [75] T. WALDMANN, A. ITURRONDOBEITIA, M. KASPER, N. GHANBARI, F. AGUESSE, E. BEKAERT, L. DANIEL, S. GENIÈS, I. GORDON, M. LÖBLE, E. DE VITO, AND M. WOHLFAHRT-MEHRENS, *Review—post-mortem analysis* of aged lithium-ion batteries: Disassembly methodology and physico-chemical analysis techniques, Journal of The Electrochemical Society, 163 (2016), pp. A2149–A2164.
- [76] C. BREITFUSS, W. SINZ, F. FEIST, G. GSTREIN, B. LICHTENEGGER, C. KNAUDER, C. ELLERSDORFER, J. MOSER, H. STEFFAN, M. STADLER, P. GOLLOB, AND V. HENNIGE, A 'microscopic' structural mechanics fe model of a lithium-ion pouch cell for quasi-static load cases, SAE International Journal of Passenger Cars - Mechanical Systems, 6 (2013), pp. 1044–1054.

- [77] L. WANG, S. YIN, AND J. XU, A detailed computational model for cylindrical lithium-ion batteries under mechanical loading: From cell deformation to shortcircuit onset, Journal of Power Sources, 413 (2019), pp. 284–292.
- [78] B. ARIEF BUDIMAN, S. RAHARDIAN, A. SAPUTRO, A. HIDAYAT, I. PU-LUNG NURPRASETIO, AND P. SAMBEGORO, Structural integrity of lithium-ion pouch battery subjected to three-point bending, Engineering Failure Analysis, 138 (2022), p. 106307.
- [79] E. SAHRAEI, E. BOSCO, B. DIXON, AND B. LAI, Microscale failure mechanisms leading to internal short circuit in li-ion batteries under complex loading scenarios, Journal of Power Sources, 319 (2016), pp. 56–65.
- [80] R. BEAUMONT, I. MASTERS, A. DAS, S. LUCAS, A. THANIKACHALAM, AND D. WILLIAMS, Methodology for developing a macro finite element model of lithium-ion pouch cells for predicting mechanical behaviour under multiple loading conditions, Energies, 14 (2021).
- [81] M. RAFFLER, A. SEVARIN, C. ELLERSDORFER, S. F. HEINDL, C. BREITFUSS, AND W. SINZ, Finite element model approach of a cylindrical lithium ion battery cell with a focus on minimization of the computational effort and short circuit prediction, Journal of Power Sources, 360 (2017), pp. 605–617.
- [82] E. SAHRAEI, J. CAMPBELL, AND T. WIERZBICKI, Modeling and short circuit detection of 18650 li-ion cells under mechanical abuse conditions, Journal of Power Sources, 220 (2012), pp. 360–372.
- [83] S. RODRIGUEZ, A. PASQUALE, K. NGUYEN, A. AMMAR, AND F. CHINESTA, A time multiscale based data-driven approach in cyclic elasto-plasticity, Computers & Structures, 295 (2024), p. 107277.
- [84] A. PASQUALE, S. RODRIGUEZ, K. NGUYEN, A. AMMAR, AND F. CHINESTA, A time multiscale decomposition in cyclic elasto-plasticity, Computers & Mathematics with Applications, 149 (2023), pp. 75–83.
- [85] C. GHNATIOS, V. CHAMPANEY, A. PASQUALE, AND F. CHINESTA, A regularized real-time integrator for data-driven control of heating channels, Computation, 10 (2022).
- [86] A. SCHMID, A. PASQUALE, C. ELLERSDORFER, M. RAFFLER, V. CHAM-PANEY, M. ZIANE, F. FEIST, AND F. CHINESTA, Mechanical characterization of li-ion cells and the calibration of numerical models using proper generalized decomposition, International Mechanical Engineering Congress and Exposition (IMECE), October 29-November 2 2023.
- [87] S. RODRIGUEZ, A. PASQUALE, N. KHANH, A. AMMAR, AND F. CHINESTA, A time multiscale based data-driven approach in cyclic elasto-plasticity, XVII International Conference on Computational Plasticity, Fundamentals and Applications (COMPLAS), September 5-7 2023.
- [88] A. PASQUALE, M. ZIANE, A. AMMAR, AND F. CHINESTA, Machine learning enhanced shell element formulation for the mechanical response of thin to thick structures, XVII International Conference on Computational Plasticity, Fundamentals and Applications (COMPLAS), September 5-7 2023.

- [89] A. PASQUALE, D. DI LORENZO, V. CHAMPANEY, F. CHINESTA, E. CUETO, AND D. BAILLARGEAT, *Towards smart city digital twins*, International Congress on Industrial and Applied Mathematics (ICIAM), August 20-25 2023.
- [90] D. DI LORENZO, V. CHAMPANEY, A. PASQUALE, AND F. CHINESTA, Models correction based on sparse identification and data assimilation, International Congress on Industrial and Applied Mathematics (ICIAM), August 20-25 2023.
- [91] A. SCHMID, C. ELLERSDORFER, M. RAFFLER, F. FEIST, A. PASQUALE, V. CHAMPANEY, S. GUÉVELOU, M. ZIANE, AND F. CHINESTA, Meta-modelling of a li-ion battery for real time prediction based on proper generalized decomposition, Advanced Materials Poster Day (Tu Graz), February 2 2023.
- [92] D. DI LORENZO, A. PASQUALE, V. CHAMPANEY, M.-J. KAZEMZADEH-PARSI, A. AMMAR, D. BAILLARGEAT, AND F. CHINESTA, The importance of wind simulation to find in real time the optimal flight trajectory in drones operations, 10th OpenFOAM Conference, November 8 2022.
- [93] V. CHAMPANEY, D. DI LORENZO, A. PASQUALE, A. AMMAR, M.-J. KAZEMZADEH-PARSI, D. BAILLARGEAT, E. CUETO, AND F. CHINESTA, *Hy*brid modelling for decision making in critical systems, IUTAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022.
- [94] A. RUNACHER, A. PASQUALE, M.-J. KAZEMZADEH-PARSI, C. JOFFRE, C. SALVAN, C. GHNATIOS, E. CUETO, AND F. CHINESTA, *Data-driven modelling of the filament shape evolution in fused filament fabrication process*, IU-TAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022.
- [95] M. JACOT, V. CHAMPANEY, A. PASQUALE, P. TRIGUERO NAVARRO, AND F. CHINESTA, Real-time structural health monitoring of aeronautics structures: a damage detection and identification approach based on a multidimensional parametric model using the sparse proper generalized decomposition coupled with optimization, IUTAM Symposium on Data-Driven Mechanics and Surrogate Modeling, October 25-28 2022.
- [96] A. PASQUALE, V. CHAMPANEY, A. AMMAR, F. CHINESTA, AND J.-L. DUVAL, *Pgd based time multiscale*, Mechanistic Machine Learning and Digital Twins for Computational Science, Engineering and Technology (MMLDT-CSET), September 26-29 2021.
- [97] E. CARRERA, F. A. FAZZOLARI, AND M. CINEFRA, *Chapter 4 fundamental of mechanics of beams, plates and shells*, in Thermal Stress Analysis of Beams, Plates and Shells, E. Carrera, F. A. Fazzolari, and M. Cinefra, eds., Academic Press, Oxford, 2017, pp. 91–116.
- [98] O. ZIENKIEWICZ, R. TAYLOR, AND D. FOX, The finite element method for solid and structural mechanics: Seventh edition, The Finite Element Method for Solid and Structural Mechanics: Seventh Edition, (2013), pp. 1–624.
- [99] E. FASANELLA AND K. JACKSON, Best practices for crash modeling and simulation, (2002).
- [100] E. A. TRÄFF, O. SIGMUND, AND N. AAGE, Topology optimization of ultra high resolution shell structures, Thin-Walled Structures, 160 (2021), p. 107349.

- [101] E. REISSNER, The Effect of Transverse Shear Deformation on the Bending of Elastic Plates, Journal of Applied Mechanics, 12 (1945), pp. A69–A77.
- [102] M. PETROLO AND E. CARRERA, Methods and guidelines for the choice of shell theories, Acta Mechanica, 231 (2020).
- [103] C. WANG, J. REDDY, AND K. LEE, Shear Deformable Beams and Plates: Relationships with Classical Solutions, Elsevier Science, 2000.
- [104] M. P. L. PARENTE, R. A. F. VALENTE, R. M. N. JORGE, R. P. R. CARDOSO, AND R. J. A. DE SOUSA, Sheet metal forming simulation using eas solid-shell finite elements, Finite Elements in Analysis and Design, 42 (2006), pp. 1137– 1149.
- [105] M. PASQUALI AND P. GAUDENZI, A geometrically exact formulation of thin laminated composite shells, Composite Structures, 180 (2017), pp. 542–549.
- [106] A. TESSLER, M. ANNETT, AND G. GENDRON, A 1,2-order plate theory accounting for three-dimensional thermoelastic deformations in thick composite and sandwich laminates, Composite Structures, 52 (2001), pp. 67–84.
- [107] E. GROUP, Virtual Performance Solution Solver Reference Manual, 08 2022.
- [108] M. PARENTE, R. FONTES VALENTE, R. NATAL JORGE, R. CARDOSO, AND R. ALVES DE SOUSA, Sheet metal forming simulation using eas solid-shell finite elements, Finite Elements in Analysis and Design, 42 (2006), pp. 1137–1149.
- [109] N. YOUNAS, H. CHALAL, AND F. ABED-MERAIM, Finite element simulation of sheet metal forming processes using non-quadratic anisotropic plasticity models and solid—shell finite elements, Procedia Manufacturing, 47 (2020), pp. 1416– 1423. 23rd International Conference on Material Forming.
- [110] M. HARNAU AND K. SCHWEIZERHOF, Artificial kinematics and simple stabilization of solid-shell elements occurring in highly constrained situations and applications in composite sheet forming simulation, Finite Elements in Analysis and Design, 42 (2006), pp. 1097–1111.
- [111] M. MAHMOUD, F. BAY, AND D. P. MUÑOZ, An efficient multiphysics solid shell based finite element approach for modeling thin sheet metal forming processes, Finite Elements in Analysis and Design, 198 (2022), p. 103645.
- [112] B. SCHÄFER, D. DÖRR, AND L. KÄRGER, Potential and challenges of a solidshell element for the macroscopic forming simulation of engineering textiles, ESAFORM 2021, (2021).
- [113] G. WEI, P. LARDEUR, AND F. DRUESNE, Solid-shell approach based on firstorder or higher-order plate and shell theories for the finite element analysis of thin to very thick structures, European Journal of Mechanics - A/Solids, 94 (2022), p. 104591.
- [114] G. COCCHETTI, M. PAGANI, AND U. PEREGO, Selective mass scaling for distorted solid-shell elements in explicit dynamics: Optimal scaling factor and stable time step estimate, International Journal for Numerical Methods in Engineering, 101 (2014).

- [115] N. EL-ABBASI AND S. MEGUID, A new shell element accounting for throughthickness deformation, Computer Methods in Applied Mechanics and Engineering, 189 (2000), pp. 841–862.
- [116] M. REZAIEE-PAJAND, A. R. MASOODI, AND E. ARABI, On the shell thicknessstretching effects using seven-parameter triangular element, European Journal of Computational Mechanics, 27 (2018), pp. 163 – 185.
- [117] A. GARG, T. MUKHOPADHYAY, M. BELARBI, H. CHALAK, A. SINGH, AND A. ZENKOUR, On accurately capturing the through-thickness variation of transverse shear and normal stresses for composite beams using fsdt coupled with gpr, Composite Structures, 305 (2023), p. 116551.
- [118] T. WILLMANN AND M. BISCHOFF, Shell models with enhanced kinematics for finite elements in sheet metal forming simulations, 12th European LS-DYNA, (2019).
- [119] J. MANTARI, A. OKTEM, AND C. GUEDES SOARES, A new higher order shear deformation theory for sandwich and composite laminated plates, Composites Part B: Engineering, 43 (2012), pp. 1489–1499.
- [120] E. RANK, A. DÜSTER, V. NÜBEL, K. PREUSCH, AND O. BRUHNS, *High or*der finite elements for shells, Computer Methods in Applied Mechanics and Engineering, 194 (2005), pp. 2494–2512.
- [121] E. CARRERA AND V. ZOZULYA, Closed-form solution for the micropolar plates: Carrera unified formulation (cuf) approach, Archive of Applied Mechanics, 91 (2021).
- [122] E. CARRERA AND V. ZOZULYA, Analytical solution for the micropolar cylindrical shell: Carrera unified formulation (cuf) approach, Continuum Mechanics and Thermodynamics, (2022), pp. 1–21.
- [123] E. CARRERA AND V. ZOZULYA, Carrera unified formulation (cuf) for the composite shells of revolution. equivalent single layer models, Mechanics of Advanced Materials and Structures, (2023), pp. 1–23.
- [124] E. CARRERA, M. CINEFRA, M. PETROLO, AND E. ZAPPINO, Finite element analysis of structures through unified formulation, Finite Element Analysis of Structures through Unified Formulation, (2014).
- [125] C. GHNATIOS, G. E. HABER, J.-L. DUVAL, M. ZIANE, AND F. CHINESTA, Artificial intelligence based space reduction of structural models, ESAFORM Conference on Material Forming [Online], (2021).
- [126] E. DE SOUZA NETO, D. PERIC, AND D. OWEN, Computational Methods for Plasticity: Theory and Applications, Wiley, 2008.
- [127] N. LIU, X. CUI, J. XIAO, J. LUA, AND N. PHAN, A simplified continuum damage mechanics based modeling strategy for cumulative fatigue damage assessment of metallic bolted joints, International Journal of Fatigue, 131 (2020), p. 105302.
- [128] M. W. JOOSTEN, C. G. DÁVILA, AND Q. YANG, Predicting fatigue damage in composites subjected to general loading conditions, Composites Part A: Applied Science and Manufacturing, 156 (2022), p. 106862.

- [129] M. D. SANGID, The physics of fatigue crack initiation, International Journal of Fatigue, 57 (2013), pp. 58–72. Fatigue and Microstructure: A special issue on recent advances.
- [130] A. AMMAR, A. ZGHAL, F. MOREL, AND F. CHINESTA, On the space-time separated representation of integral linear viscoelastic models, Comptes Rendus Mécanique, 343 (2015), pp. 247–263.
- [131] S. T. T. NGUYEN, S. CASTAGNET, AND J.-C. GRANDIDIER, Nonlinear viscoelastic contribution to the cyclic accommodation of high density polyethylene in tension: Experiments and modeling, International Journal of Fatigue, 55 (2013), pp. 166–177.
- [132] J.-C. G. MOHAMMAD HAMMOUD, MARIANNE BERINGHIER, A reduced simulation applied to the viscoelastic fatigue of polymers, Comptes Rendus Mécanique, 4349 (2014), pp. 671–746.
- [133] Y. SUN, W. ZENG, Y. ZHAO, X. ZHANG, Y. SHU, AND Y. ZHOU, Modeling constitutive relationship of ti40 alloy using artificial neural network, Materials & Design, 32 (2011), pp. 1537–1541.
- [134] G. TEICHERT, A. NATARAJAN, A. VAN DER VEN, AND K. GARIKIPATI, Machine learning materials physics: Integrable deep neural networks enable scale bridging by learning free energy functions, Computer Methods in Applied Mechanics and Engineering, 353 (2019), pp. 201–216.
- [135] Y. ZHANG, Z. WEN, H. PEI, J. WANG, Z. LI, AND Z. YUE, Equivalent method of evaluating mechanical properties of perforated ni-based single crystal plates using artificial neural networks, Computer Methods in Applied Mechanics and Engineering, 360 (2020), p. 112725.
- [136] J. SIMO AND R. TAYLOR, Consistent tangent operators for rate-independent elastoplasticity, Computer Methods in Applied Mechanics and Engineering, 48 (1985), pp. 101–118.
- [137] L. XU, X. NIE, J. FAN, M. TAO, AND R. DING, Cyclic hardening and softening behavior of the low yield point steel bly160: Experimental response and constitutive modeling, International Journal of Plasticity, 78 (2016), pp. 44–63.
- [138] C. WANG, L. YAN XU, AND J. SHENG FAN, A general deep learning framework for history-dependent response prediction based on ua-seq2seq model, Computer Methods in Applied Mechanics and Engineering, 372 (2020), p. 113357.
- [139] P. LADEVÈZE, Sur une famille d'algorithmes en mécanique des structures, Comptes-rendus des séances de l'Académie des sciences. Série 2, Mécaniquephysique, chimie, sciences de l'univers, sciences de la terre, 300 (1985), pp. 41– 44.
- [140] J.-Y. COGNARD AND P. LADEVÈZE, A large time increment approach for cyclic viscoplasticity, International Journal of plasticity, 9 (1993), pp. 141–157.
- [141] M. ARZT AND P. LADEVÈZE, Approche des phénomenes cycliques par la méthode à grand incrément de temps, PhD thesis, Cachan, Ecole normale supérieure, 1994.

- [142] M. BHATTACHARYYA, A. FAU, U. NACKENHORST, D. NÉRON, AND P. LADE-VÈZE, A model reduction technique in space and time for fatigue simulation, in Multiscale modeling of heterogeneous structures, Springer, 2018, pp. 183–203.
- [143] M. BHATTACHARYYA, A. FAU, U. NACKENHORST, D. NÉRON, AND P. LADE-VÈZE, A latin-based model reduction approach for the simulation of cycling damage, Computational Mechanics, 62 (2018), pp. 725–743.
- [144] M. BHATTACHARYYA, A. FAU, U. NACKENHORST, D. NÉRON, AND P. LADE-VÈZE, A multi-temporal scale model reduction approach for the computation of fatigue damage, Computer Methods in Applied Mechanics and Engineering, 340 (2018), pp. 630–656.
- [145] M. BHATTACHARYYA, A. FAU, R. DESMORAT, S. ALAMEDDIN, D. NÉRON, P. LADEVÈZE, AND U. NACKENHORST, A kinetic two-scale damage model for high-cycle fatigue simulation using multi-temporal latin framework, European Journal of Mechanics-A/Solids, 77 (2019), p. 103808.
- [146] M. CAPALDO, A new approximation framework for PGD-based nonlinear solvers, PhD thesis, 2015.
- [147] M. CAPALDO, P.-A. GUIDAULT, D. NÉRON, AND P. LADEVÈZE, The reference point method, a hyperreduction technique: Application to pgd-based nonlinear model reduction, Computer Methods in Applied Mechanics and Engineering, 322 (2017), pp. 483–514.
- [148] R. IBÁÑEZ, A. AMMAR, E. CUETO, A. HUERTA, J.-L. DUVAL, AND F. CHINESTA, Multiscale proper generalized decomposition based on the partition of unity, International Journal for Numerical Methods in Engineering, 120 (2019), pp. 727–747.
- [149] O. SALLY, F. LAURIN, C. JULIEN, R. DESMORAT, AND F. BOUILLON, An efficient computational strategy of cycle-jumps dedicated to fatigue of composite structures, International Journal of Fatigue, 135 (2020), p. 105500.
- [150] M. BHATTACHARYYA, A. FAU, U. NACKENHORST, D. NÉRON, AND P. LADE-VEZE, A Model Reduction Technique in Space and Time for Fatigue Simulation, 01 2018, pp. 183–203.
- [151] M. BHATTACHARYYA, A. FAU, U. NACKENHORST, D. NÉRON, AND P. LADE-VÈZE, A multi-temporal scale model reduction approach for the computation of fatigue damage, Computer Methods in Applied Mechanics and Engineering, 340 (2018), pp. 630–656.
- [152] D. COJOCARU AND A. KARLSSON, A simple numerical method of cycle jumps for cyclically loaded structures, International Journal of Fatigue, 28 (2006), pp. 1677–1689.
- [153] J. SIMO AND T. HUGHES, Computational Inelasticity, Interdisciplinary Applied Mathematics, Springer New York, 2006.
- [154] K. HASHIGUCHI AND Y. YAMAKAWA, Introduction to Finite Strain Theory for Continuum Elasto-Plasticity, Wiley, 2012.

- [155] D. NÉRON AND P. LADEVÈZE, Proper generalized decomposition for multiscale and multiphysics problems, Archives of Computational Methods in Engineering, 17 (2010), pp. 351–372.
- [156] M. A. NASRI, C. ROBERT, A. AMMAR, S. EL AREM, AND F. MOREL, Proper generalized decomposition (pgd) for the numerical simulation of polycrystalline aggregates under cyclic loading, Comptes Rendus Mécanique, 346 (2018), pp. 132–151.
- [157] J.-M. BERGHEAU, S. ZUCHIATTI, J.-C. ROUX, E. FEULVARCH, S. TISSOT, AND G. PERRIN, The proper generalized decomposition as a space-time integrator for elastoplastic problems, Comptes Rendus Mecanique, 344 (2016), pp. 759– 768.
- [158] T. ARJOUNE, B. MARKERT, AND F. BAMER, Non-incremental response evaluation in geometrically nonlinear structural dynamics using a space-time stiffness operator, Computational Mechanics, 70 (2022).
- [159] H. JAHED AND A. ROOSTAEI, Cyclic Plasticity of Metals: Modeling Fundamentals and Applications, Elsevier Series on Plasticity of Materials, Elsevier Science, 2021.
- [160] F. CHINESTA, A. LEYGUE, F. BORDEU, J. V. AGUADO, E. CUETO, D. GON-ZALEZ, I. ALFARO, A. AMMAR, AND A. HUERTA, *Pgd-based computational* vademecum for efficient design, optimization and control, Archives of Computational Methods in Engineering, 20 (2013).
- [161] 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, 347 (2019), pp. 445–454.
- [162] M.-J. KAZEMZADEH-PARSI, A. AMMAR, J. DUVAL, AND F. CHINESTA, Enhanced parametric shape descriptions in pgd-based space separated representations, Advanced Modeling and Simulation in Engineering Sciences, 8 (2021).
- [163] F. CHINESTA, A. AMMAR, AND E. CUETO, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Archives of Computational Methods in Engineering, 17 (2010), pp. 327– 350.
- [164] P. LADEVÈZE, Pgd in linear and nonlinear computational solid mechanics, in Separated representations and PGD-based model reduction, Springer, 2014, pp. 91–152.
- [165] 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, 255 (2013), pp. 67 – 88.
- [166] M.-J. KAZEMZADEH-PARSI, F. CHINESTA, AND A. AMMAR, Proper generalized decomposition for parametric study and material distribution design of multidirectional functionally graded plates based on 3d elasticity solution, Materials, 14 (2021).

- [167] M.-J. KAZEMZADEH-PARSI, A. AMMAR, AND F. CHINESTA, Domain decomposition involving subdomain separable space representations for solving parametric problems in complex geometries, Advanced Modeling and Simulation in Engineering Sciences, 9 (2022).
- [168] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, A multilinear singular value decomposition, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1253–1278.
- [169] R. BADEAU AND R. BOYER, Fast multilinear singular value decomposition for structured tensors, SIAM J. Matrix Analysis Applications, 30 (2008), pp. 1008– 1021.
- [170] E. SAMANIEGO, C. ANITESCU, S. GOSWAMI, V. NGUYEN-THANH, H. GUO, K. HAMDIA, X. ZHUANG, AND T. RABCZUK, An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications, Computer Methods in Applied Mechanics and Engineering, 362 (2020), p. 112790.
- [171] S. TIRUNAGARI, S. KOUCHAKI, N. POH, M. BOBER, AND D. WINDRIDGE, Dynamic Mode Decomposition for Univariate Time Series: Analysing Trends and Forecasting. working paper or preprint, Feb. 2017.
- [172] D. DYLEWSKY, D. BARAJAS-SOLANO, T. MA, A. TARTAKOVSKY, AND J. KUTZ, Stochastically forced ensemble dynamic mode decomposition for forecasting and analysis of near-periodic systems, IEEE Access, 10 (2022), pp. 1–1.
- [173] J. LIEW, T. GÖÇMEN, W. H. LIO, AND G. C. LARSEN, Streaming dynamic mode decomposition for short-term forecasting in wind farms, Wind Energy, 25 (2022), pp. 719–734.
- [174] A. SERANI, P. DRAGONE, F. STERN, AND M. DIEZ, On the use of dynamic mode decomposition for time-series forecasting of ships operating in waves, Ocean Engineering, 267 (2023), p. 113235.
- [175] P. SCHMID AND J. SESTERHENN, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656 (2008).
- [176] R. WANG, Y. DONG, S. O. ARIK, AND R. YU, Koopman neural operator forecaster for time-series with temporal distributional shifts, in International Conference on Learning Representations, 2023.
- [177] A. DOTTO, D. LENGANI, D. SIMONI, AND A. TACCHELLA, Dynamic mode decomposition and koopman spectral analysis of boundary layer separation-induced transition, Physics of Fluids, 33 (2021), p. 104104.
- [178] Z. WU, S. L. BRUNTON, AND S. REVZEN, Challenges in dynamic mode decomposition, Journal of The Royal Society Interface, 18 (2021), p. 20210686.
- [179] S. LE CLAINCHE AND J. M. VEGA, *Higher order dynamic mode decomposition*, SIAM Journal on Applied Dynamical Systems, 16 (2017), pp. 882–925.
- [180] J. M. VEGA AND S. LE CLAINCHE, *Higher Order Dynamic Mode Decomposition* and its Applications, Academic, 2020.

- [181] R. EVERSON AND L. SIROVICH, Karhunen-loève procedure for gappy data, JOSA A, 12 (1995).
- [182] K. WILLCOX, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, Computers & Fluids, 35 (2006), pp. 208–226.
- [183] D. RYCKELYNCK AND D. MISSOUM BENZIANE, Multi-level a priori hyperreduction of mechanical models involving internal variables, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 1134–1142.
- [184] B. CHUN, J. JINN, AND J. LEE, Modeling the bauschinger effect for sheet metals, part i: theory, International Journal of Plasticity, 18 (2002), pp. 571–595.
- [185] V. VELAY, G. BERNHART, AND L. PENAZZI, Cyclic behavior modeling of a tempered martensitic hot work tool steel, International Journal of Plasticity, 22 (2006), pp. 459–496.
- [186] G. KANG, Finite element implementation of visco-plastic constitutive model with strain-range-dependent cyclic hardening, Communications in Numerical Methods in Engineering, 22 (2006), pp. 137–153.
- [187] P. K. V. NUKALA, A return mapping algorithm for cyclic viscoplastic constitutive models, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 148–178.
- [188] B. PEHERSTORFER, Z. DRMAČ, AND S. GUGERCIN, Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points, SIAM Journal on Scientific Computing, 42 (2020), pp. A2837–A2864.
- [189] J. S. HALE, E. SCHENONE, D. BAROLI, L. A. BEEX, AND S. P. BORDAS, A hyper-reduction method using adaptivity to cut the assembly costs of reduced order models, Computer Methods in Applied Mechanics and Engineering, 380 (2021), p. 113723.
- [190] F. MOZAFARI, P. THAMBURAJA, A. SRINIVASA, AND S. ABDULLAH, Fatigue life prediction under variable amplitude loading using a microplasticity-based constitutive model, International Journal of Fatigue, 134 (2020), p. 105477.
- [191] P. BENNER, M. OHLBERGER, A. COHEN, AND K. WILLCOX, Model Reduction and Approximation, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.
- [192] G. ROZZA, M. HESS, G. STABILE, M. TEZZELE, F. BALLARIN, C. GRÄSSLE, M. HINZE, S. VOLKWEIN, F. CHINESTA, P. LADEVEZE, Y. MADAY, A. PA-TERA, C. FARHAT, S. GRIMBERG, A. MANZONI, A. QUARTERONI, A. BUHR, L. IAPICHINO, AND J. KUTZ, Volume 2 Snapshot-Based Methods and Algorithms, De Gruyter, 12 2020.
- [193] G. ROZZA, M. HESS, G. STABILE, M. TEZZELE, F. BALLARIN, C. GRÄSSLE, M. HINZE, S. VOLKWEIN, F. CHINESTA, P. LADEVEZE, Y. MADAY, A. PA-TERA, C. FARHAT, S. GRIMBERG, A. MANZONI, A. QUARTERONI, A. BUHR, L. IAPICHINO, AND J. KUTZ, *Model Order Reduction: Volume 3: Applications*, De Gruyter, 2020.

- [194] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, SpringerBriefs in Mathematics, Springer International Publishing, Jan. 2015. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-22470-1.
- [195] A. QUARTERONI, A. MANZONI, AND F. NEGRI, Reduced basis methods for partial differential equations: An introduction, 01 2015.
- [196] M. OHLBERGER AND S. RAVE, Reduced basis methods: Success, limitations and future challenges, arXiv: Numerical Analysis, (2015).
- [197] K. URBAN, The Reduced Basis Method in Space and Time: Challenges, Limits and Perspectives, Springer Nature Switzerland, Cham, 2023, pp. 1–72.
- [198] D. BORZACCHIELLO, J. V. AGUADO, AND F. CHINESTA, Non-intrusive sparse subspace learning for parametrized problems, Archives of Computational Methods in Engineering, 26 (2017).
- [199] R. IBÁÑEZ PINILLO, E. ABISSET-CHAVANNE, A. AMMAR, D. GONZÁLEZ, E. CUETO, A. HUERTA, J. DUVAL, AND F. CHINESTA, A multidimensional data-driven sparse identification technique: The sparse proper generalized decomposition, Complexity, 2018 (2018), pp. 1–11.
- [200] A. SANCARLOS, V. CHAMPANEY, E. CUETO, AND F. CHINESTA, Regularized regressions for parametric models based on separated representations, Advanced Modeling and Simulation in Engineering Sciences, 10 (2023).
- [201] V. CHAMPANEY, F. CHINESTA, AND E. CUETO, Engineering empowered by physics-based and data-driven hybrid models: A methodological overview, International Journal of Material Forming, 15 (2022).
- [202] N. DEMO, M. TEZZELE, AND G. ROZZA, A non-intrusive approach for the reconstruction of pod modal coefficients through active subspaces, Comptes Rendus Mécanique, 347 (2019), pp. 873–881. Data-Based Engineering Science and Technology.
- [203] D. AMSALLEM AND C. FARHAT, Interpolation method for adapting reducedorder models and application to aeroelasticity, Aiaa Journal - AIAA J, 46 (2008), pp. 1803–1813.
- [204] O. FRIDERIKOS, E. BARANGER, M. OLIVE, AND D. NÉRON, On the stability of pod basis interpolation via grassmann manifolds for parametric model order reduction in hyperelasticity, 2020.
- [205] K. VLACHAS, K. TATSIS, K. AGATHOS, A. R. BRINK, AND E. CHATZI, A local basis approximation approach for nonlinear parametric model order reduction, Journal of Sound and Vibration, 502 (2021), p. 116055.
- [206] R. ZIMMERMANN, Manifold interpolation and model reduction, 2022.
- [207] S. TORREGROSA, V. CHAMPANEY, A. AMMAR, V. HERBERT, AND F. CHINESTA, Surrogate parametric metamodel based on optimal transport, Mathematics and Computers in Simulation, 194 (2021).
- [208] A. BUHR, L. IAPICHINO, M. OHLBERGER, S. RAVE, F. SCHINDLER, AND K. SMETANA, Localized model reduction for parameterized problems, 2019.

- [209] C. FARHAT AND F.-X. ROUX, A method of finite element tearing and interconnecting and its parallel solution algorithm, International Journal for Numerical Methods in Engineering, 32 (1991), pp. 1205–1227.
- [210] C. FARHAT AND F.-X. ROUX, An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, SIAM Journal on Scientific and Statistical Computing, 13 (1992), pp. 379–396.
- [211] C. FARHAT, J. MANDEL, AND F. X. ROUX, Optimal convergence properties of the feti domain decomposition method, Computer Methods in Applied Mechanics and Engineering, 115 (1994), pp. 365–385.
- [212] D. HUYNH, D. KNEZEVIC, AND A. PATERA, A static condensation reduced basis element method: approximation and a posteriori error estimation, European Series in Applied and Industrial Mathematics (ESAIM): Mathematical Modelling and Numerical Analysis, 47 (2013).
- [213] L. IAPICHINO, A. QUARTERONI, AND G. ROZZA, Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries, Computers & Mathematics with Applications, 71 (2016), pp. 408– 430.
- [214] E. ZAPPON, A. MANZONI, P. GERVASIO, AND A. QUARTERONI, A reduced order model for domain decompositions with non-conforming interfaces, 06 2022.
- [215] X. ZHAO, M. H. DAO, AND Q. T. LE, Digital twining of an offshore wind turbine on a monopile using reduced-order modelling approach, Renewable Energy, 206 (2023), pp. 531–551.
- [216] D. A. WHITE, J. KUDO, K. SWARTZ, D. A. TORTORELLI, AND S. WATTS, A reduced order model approach for finite element analysis of cellular structures, Finite Elements in Analysis and Design, 214 (2023), p. 103855.
- [217] N. IYENGAR, D. RAJARAM, K. DECKER, C. PERRON, AND D. N. MAVRIS, Nonlinear Reduced Order Modeling using Domain Decomposition.
- [218] T. R. F. PHILLIPS, C. E. HEANEY, B. S. TOLLIT, P. N. SMITH, AND C. C. PAIN, Reduced-order modelling with domain decomposition applied to multigroup neutron transport, Energies, 14 (2021).
- [219] D. AMSALLEM, M. ZAHR, AND C. FARHAT, Nonlinear model order reduction based on local reduced-order bases, International Journal for Numerical Methods in Engineering, 92 (2012), pp. 891–916.
- [220] C. HOANG, Y. CHOI, AND K. CARLBERG, Domain-decomposition least-squares petrov-galerkin (dd-lspg) nonlinear model reduction, Computer Methods in Applied Mechanics and Engineering, 384 (2021), p. 113997.
- [221] C. HUANG, K. DURAISAMY, AND C. MERKLE, Component-based reduced order modeling of large-scale complex systems, Frontiers in Physics, 10 (2022).
- [222] S. ANDERSON, C. WHITE, AND C. FARHAT, Space-local reduced-order bases for accelerating reduced-order models through sparsity, International Journal for Numerical Methods in Engineering, 124 (2023), pp. 1646–1671.

- [223] A. N. DIAZ, Y. CHOI, AND M. HEINKENSCHLOSS, A fast and accurate domaindecomposition nonlinear manifold reduced order model, 2023.
- [224] W. WU, X. FENG, AND H. XU, Improved deep neural networks with domain decomposition in solving partial differential equations, Journal of Scientific Computing, 93 (2022).
- [225] J. BARNETT, I. TEZAUR, AND A. MOTA, The schwarz alternating method for the seamless coupling of nonlinear reduced order models and full order models, 10 2022.
- [226] A. IOLLO, G. SAMBATARO, AND T. TADDEI, A one-shot overlapping schwarz method for component-based model reduction: application to nonlinear elasticity, Computer Methods in Applied Mechanics and Engineering, 404 (2023), p. 115786.
- [227] A. CASTRO, P. KUBERRY, I. TEZAUR, AND P. BOCHEV, A novel partitioned approach for reduced order model – finite element model (rom-fem) and rom-rom coupling, 06 2022.
- [228] A. CASTRO, P. BOCHEV, P. KUBERRY, AND I. TEZAUR, Explicit synchronous partitioned scheme for coupled reduced order models based on composite reduced bases, 06 2023.
- [229] M.-J. KAZEMZADEH-PARSI, A. AMMAR, AND F. CHINESTA, Domain decomposition involving subdomain separable space representations for solving parametric problems in complex geometries, Advanced Modeling and Simulation in Engineering Sciences, 9 (2022).
- [230] B. SCHÖLKOPF, A. SMOLA, AND K.-R. MÜLLER, Kernel principal component analysis, in Artificial Neural Networks—ICANN'97: 7th International Conference Lausanne, Switzerland, October 8–10, 1997 Proceeedings, Springer, 2005, pp. 583–588.
- [231] S. T. ROWEIS AND L. K. SAUL, Nonlinear dimensionality reduction by locally linear embedding, science, 290 (2000), pp. 2323–2326.
- [232] A. AMMAR, F. CHINESTA, AND A. FALCO, On the convergence of a greedy rankone update algorithm for a class of linear systems, Archives of Computational Methods in Engineering, 17 (2010), pp. 473–486.
- [233] 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, 30 (2008), pp. 1084–1127.
- [234] L. PIEGL AND W. TILLER, *The NURBS Book*, Monographs in Visual Communication, Springer Berlin Heidelberg, 1996.
- [235] F. CHINESTA, R. KEUNINGS, AND A. LEYGUE, The proper generalized decomposition for advanced numerical simulations. A primer, 01 2014.
- [236] D. GONZÁLEZ, A. AMMAR, F. CHINESTA, AND E. CUETO, Recent advances in the use of separated representations, International Journal for Numerical Methods in Engineering, 81 (2009), pp. 637 – 659.

LIST OF FIGURES

1	Exemples de structures minces dans la caisse en blanc (avec l'aimable autorisation du groupe ESI).	xi
2	Exemple d'une structure multiparamétrique dans l'industrie automobile (avec l'aimable autorisation du groupe Hyundai Motor)	xiii
3	Décomposition d'une grille de calcul 3D dans une géométrie en forme de plaque	xvi
4	Décomposition d'un maillage temporel fin à une échelle en un maillage à plusieurs échelles	xvii
5	Décomposition d'un problème paramétrique unique \mathcal{P} en sous-problèmes par composant \mathcal{P}_p .	xviii
1.1	Examples of thin structures in the body in white (courtesy of ESI group)	. 3
1.2	Example of a multi-parametric structure in automotive (courtesy of Hyundai Motor Group).	5
1.3	Decomposition of a 3D computational grid in a plate-like geometry	8
1.4	Decomposition of a single-scale fine time grid into a multi-scale one. $% \mathcal{A}^{(n)}$.	9
1.5	Decomposition of a single parametric problem \mathcal{P} into by-component subproblems \mathcal{P}_p .	10
1.6	Vehicle frontal structure under study (courtesy of Hyundai Motor Group)	. 12
1.7	Plasticity curve (σ, ε) and location of sampled points	13
1.8	Multi-PGD procedure	14
1.9	Various length scales occurring in EV crash simulation (courtesy of ESI Group).	14
1.10	Detailed cell model structure (courtesy of ESI Group)	15
1.11	Macroscopic modeling approach (courtesy of Vehicle Safety Institute).	16
1.12	Isometric view of four different test configurations (courtesy of Vehicle Safety Institute).	16
2.1	Shell-to-solid remeshing (courtesy of ESI Group).	21
2.2	Example of a shell-like structure	22
2.3	Kirchhoff versus Reissner–Mindlin theory.	24
2.4	Illustration of a 2D shell-elements based simulation	24
2.5	Illustration of a 3D finite-elements based simulation	25
2.6	Association of solid-based mesh to shell-based one	26

2.7	Implementation prototype in ESI VPS – Creation of a dynamic library containing a machine learning model framework.	27
2.8	Illustration of a 2D enhanced simulation	30
2.9	Geometry and loading conditions (test case 1).	32
2.10	3D displacement contours (test case 1).	32
2.11	Out-of-plane displacement profiles (test case 1)	32
2.12	Out-of-plane stress profiles (test case 1)	33
2.13	Geometry and loading conditions (test case 2)	33
2.14	3D displacement contours (test case 2)	33
2.15	Out-of-plane displacement profiles (test case 2)	34
2.16	Out-of-plane stress profiles (test case 2)	34
2.17	Geometry and loading conditions (test case 3)	34
2.18	3D displacement contours (test case 3).	35
2.19	Plastic strain and Von Mises stress contours (test case 3). \ldots	35
2.20	Out-of-plane plastic strain profiles (test case 3)	35
2.21	Out-of-plane displacement profiles (test case 3)	36
2.22	Out-of-plane stress profiles (test case 3)	36
2.23	Geometry and loading conditions (test case 4)	36
2.24	3D displacement contours (test case 4)	37
2.25	Plastic strain and Von Mises stress contours (test case 4). \ldots	37
2.26	Out-of-plane plastic strain profiles (test case 4)	37
2.27	Out-of-plane displacement profiles (test case 4)	37
2.28	Out-of-plane stress profiles (test case 4)	38
2.29	Geometry and loading conditions (test case 5)	38
2.30	3D displacement contours (test case 5)	38
2.31	Plastic strain and Von Mises stress contours (test case 5). \ldots .	39
2.32	Out-of-plane plastic strain profiles (test case 5)	39
2.33	Out-of-plane displacement profiles (test case 5)	39
2.34	Out-of-plane stress profiles (test case 5)	39
3.1	Mechanical problem under study	45
3.2	PGD-based solving scheme for elasto-plasticity	50
3.3	Discretized geometry (left) and imposed displacement (right)	53
3.4	Displacement field (left) and isotropic hardening (right) at final time T_f .	53
3.5	Stress-displacement curve (left) and hysteresis loop (right) in (0,0). Red line: FE, black dashed line: PGD	53
3.6	First four normalized PGD modes.	53
3.7	Microscale and macroscale time discretization.	54
3.8	PGD time modes $\{U_k^t(t)\}_{k=1}^4$ with 60 cycles	54
3.9	First four micro-macro modes of the multi-time decomposition of U_2^t .	55

3.10	Discretized geometry (left) and imposed displacement (right)	56
3.11	Displacement field (left) and isotropic hardening (right) at final time T_f .	56
3.12	Stress-displacement curve (left) and hysteresis loop (right) in (0,0). Red line: FE, black dashed line: PGD	56
3.13	First four normalized PGD modes	57
3.14	First four micro-macro modes of the multi-time decomposition of U_2^t .	57
3.15	PGD solving scheme for elasto-plasticity	63
3.16	Workflow of the time-multiscale based data-driven approach	64
3.17	Macrotime forecast and response prediction through the microtime pat- ters	66
3.18	Transient zone and stabilized response	67
3.19	Data-driven MT-PGD solving scheme for elasto-plasticity.	70
3.20	Discretized geometry (left) and imposed displacement (right)	72
3.21	Displacement field (left) and isotropic hardening (right) at $T_f = T_K$.	72
3.22	Components of the plastic strain tensor $\boldsymbol{\varepsilon}^p = (\varepsilon_{11}^p, \varepsilon_{12}^p, \varepsilon_{22}^p)$ at $T_f = T_K$.	72
3.23	First four normalized POD time modes of $\boldsymbol{\varepsilon}^{p,\mathrm{POD}}(\boldsymbol{x},t)$	73
3.24	First three normalized MT-PGD time modes of $\boldsymbol{\varepsilon}^{p,\mathrm{MT-PGD}}(\boldsymbol{x},t)$	73
3.25	First three normalized MT-PGD space modes of $\boldsymbol{\varepsilon}^{p,\mathrm{MT-PGD}}(\boldsymbol{x},t)$	74
3.26	HODMD-based prediction of the macrotime modes of $\boldsymbol{\varepsilon}^{p,\mathrm{MT}\text{-}\mathrm{PGD}}(\boldsymbol{x},t)$.	74
3.27	Reference solution –blue line– compared with the predicted response $\hat{\psi}(t)$ –red line– and with the predicted-corrected one $\hat{\psi}^*(t)$ –green line–, in the spatial location $\boldsymbol{x}_c = (0, 0)$.	75
3.28	Cyclic displacement with linearly increasing average	75
3.29	HODMD-based modes prediction of the macrotime modes (left) and reference solution –blue line– compared with the predicted-corrected response $\hat{\psi}^*(t)$ –green line– in the spatial location $\boldsymbol{x}_{+} = (0, 0)$ (right)	75
3 30	Discretized geometry (left) and imposed displacement (right).	76
3.31	Displacement field (left) and isotropic hardening (right) at $T_F = T_K$	77
3.32	Components of the plastic strain tensor $\varepsilon^p = (\varepsilon_{12}^p, \varepsilon_{12}^p, \varepsilon_{22}^p)$ at $T_f = T_K$.	 77
3.33	First four normalized POD time modes of $\varepsilon^{p,\text{POD}}(x,t)$.	77
3.34	First three normalized MT-PGD time modes of $\boldsymbol{\varepsilon}^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.	78
3.35	First three normalized MT-PGD space modes of $\boldsymbol{\varepsilon}^{p,\text{MT-PGD}}(\boldsymbol{x},t)$.	78
3.36	HODMD-based prediction of the macrotime modes of $\varepsilon^{p,\text{MT-PGD}}(x,t)$.	78
3.37	Reference solution –blue line– compared with the predicted response $\hat{\psi}(t)$ –red line– and with the predicted-corrected one $\hat{\psi}^*(t)$ –green line–, in the spatial location $\boldsymbol{x}_c = (0, 0.015)$.	79
4.1	Modularisation of the parametric problem \mathcal{P} defined on domain Ω and parameter vector \boldsymbol{p} into a set of some pre-solved sub-problems \mathcal{P}_i de-	

4.2	Using the NURBS-PGD technique to map the parametric modules (sub- problems) into a regular (hyper-cubic) computational space and then	
	solving it	84
4.3	Two modules sharing topology and parameters	85
4.4	Physical domain parametrized by one of its control points (white dot) and several shapes obtained by moving its location.	86
4.5	PGD computational domain after separation of space and parameters.	88
4.6	Example of PGD solution over a parametrized geometry	88
4.7	Large structure composed by P parts linked by S interfaces	89
4.8	Two parts sharing the same interface γ_s	89
4.9	Parametric sub-components	89
4.10	Parametric sub-components with parametrized boundary condition	90
4.11	Compatibility of interface between two parts	91
4.12	Steady state conduction problem set-up	93
4.13	Parametric patches decomposition	94
4.14	Reference sub-problem for \mathcal{P}_1 and \mathcal{P}_3	94
4.15	12 snapshots of parametric sub-solutions.	95
4.16	6 snapshots of the assembled parametric solution	95
4.17	Plate problem set-up	96
4.18	From modules to a reference parametric patch.	97
4.19	Plate bending from different isometric views.	97

LIST OF TABLES

2.1	Operator B_i , for $i = 1, \ldots, 4$.	29
2.2	Simulation parameters (test case 1)	31
2.3	Simulation parameters (test case 2)	33
2.4	Simulation parameters (test case 3)	34
2.5	Simulation parameters (test case 4)	36
2.6	Simulation parameters (test case 5)	38
3.1	Material parameters for steel dog-bone shaped specimen	71
3.2	Material parameters for steel plate with hole	76
4.1	Problem parameters and ranges	96

NOVEL SEPARATED REPRESENTATIONS FOR CHALLENGING INDUSTRIAL APPLICATIONS: SPACE, TIME AND PARAMETERS

ABSTRACT

Every transient problem in continuum mechanics is characterized by three variables: space, time and parameters. The space defines the physical domain, enabling the definition of diverse systems. The time captures dynamic processes, allowing for transient behavior analysis. The parameters control system and modeling characteristics. Together, these elements drive the accuracy and relevance of computational science, making them essential for understanding and predicting real-world phenomena.

The complexity arising from managing space, time, and parameters in numerical simulations can be particularly challenging in some scenarios. This is the case, for instance, when dealing with thin structures, small time steps combined with long time intervals, and a high number of parameters over large domains.

The numerical simulation of three-dimensional models in thin geometries presents important challenges since maintaining the mesh granularity proportional to the thickness dimension requires an impractical number of elements for the entire structure. This issue is currently encountered in automotive industry when considering vehicle crash simulations, where most of the components are thin structures.

When time multiscale behaviours occur, standard discretization techniques are constraint to mesh up to the finest scale to predict accurately the response of the system. This results in a prohibitive computational when the phenomena must be observed over a long duration. This occurs, for instance, in material science when dealing with fatigue damage assessments and cyclic visco-elasto-plastic fatigue problems.

A large number of parameters increases the dimensionality of the parameter space exponentially, making its exploration computationally intensive. The data generated from numerous simulations can be difficult to manage and advanced meta-modeling techniques are required. This typically happens in optimal design problems of multicomponent parametric structures.

To address these challenges, it is essential to strike a balance between accuracy and computational efficiency, requiring ad-hoc advanced developments. In this thesis the three challenges are separately addressed via novel separation-based techniques.

Nouvelles représentations séparées pour des applications industrielles complexes : espace, temps et paramètres

Résumé

Chaque problème transitoire en mécanique des milieux continus est caractérisé par trois variables : l'espace, le temps et les paramètres. L'espace définit le domaine physique, permettant la description de divers systèmes. Le temps capture les processus dynamiques, autorisant l'analyse des comportements transitoires. Les paramètres influent sur les caractéristiques des systèmes et de la modélisation. Ensemble, ces éléments influent sur l'exactitude et la pertinence de la science computationnelle, les rendant essentiels pour la compréhension et la prédiction des phénomènes réels.

La gestion de l'espace, du temps et des paramètres dans les simulations numériques est particulièrement impactée dans certains scenarios. Cela est le cas, par exemple, lorsqu'il s'agit de structures minces, de petits pas de temps combinés à de longs intervalles, et d'un grand nombre de paramètres sur de vastes domaines.

La simulation numérique de modèles tridimensionnels dans des géométries minces présente d'importants défis, car maintenir la granularité du maillage proportionnelle à la dimension de l'épaisseur nécessite un nombre impraticable d'éléments pour toute la structure. Cela survient souvent dans l'industrie automobile lors de simulations de crash, où la plupart des composants sont des structures minces.

Lorsque des comportements multiscales dans le temps surviennent, les techniques de discrétisation standard sont contraintes de mailler jusqu'à l'échelle la plus fine pour prédire avec précision la réponse du système. Cela entraîne un coût computationnel prohibitif lorsque les phénomènes sont observés sur une longue durée, comme c'est le cas en science des matériaux lors de l'évaluation des endommagements par fatigue.

Un grand nombre de paramètres augmente de manière exponentielle la dimension de l'espace paramètrique, limitant son exploration. Les données générées par de nombreuses simulations peuvent être difficiles à gérer et des techniques avancées de métamodélisation sont nécessaires. Cela se produit généralement dans des problèmes de conception optimale de structures paramétriques à plusieurs composants.

Pour relever ces défis, il est essentiel de trouver un équilibre entre la précision et l'efficacité computationnelle, nécessitant des développements avancés. Dans cette thèse, les trois défis sont abordés via des nouvelles techniques basées sur les représentations séparéés.