Advanced physics-based and data-driven strategies

Rubén Ibáñez Pinillo

Écolé Centrale de Nantes
Nantes, France

Universitat Politècnica de Catalunya
Barcelona, Spain

P H D   T H E S I S

Advanced physics-based and data-driven strategies

Dissertation presented in partial fulfilment of the requirements for the degree of

PhD of the Universitat Politècnica de Catalunya
Doctor de la Universitat Politècnica de Catalunya

PhD of the École Centrale de Nantes
Docteur de l'École Centrale de Nantes

by
Rubén IBÁÑEZ PINILLO

September 2019

President:	Michel Visonneau	-	Ecole Centrale de Nantes
Jury members:	Pedro Diez	-	Universitat Politècnica de Catalunya
	Benjamin Klusemann	-	Leuphana Universität Lünenburg
	Ruben Sevilla	-	Swansea University
	Antonio Gil	-	Swansea University
	David Gonzalez	-	Universidad de Zaragoza
Main Advisors:	Francisco Chinesta	-	École National Supérieure des Arts et Métiers
	Antonio Huerta	-	Universitat Politècnica de Catalunya
Co-Advisors:	Emmanuelle Abisset-Chavanne	-	École National Supérieure des Arts et Métiers
	Elías Cueto	-	Universidad de Zaragoza
	Jean-Louis Duval	-	ESI Group

Abstract

Simulation Based Engineering Science (SBES) has brought major improvements in optimization, control and inverse analysis, all leading to a deeper understanding in many processes occuring in the real world. These noticeable breakthroughts are present in a vast variety of sectors such as aeronautic or automotive industries, mobile telecommunications or healthcare among many other fields. Nevertheless, SBES is currently confronting several difficulties to provide accurate results in complex industrial problems. Apart from the high computational costs associated with industrial applications, the errors introduced by constitutive modeling become more and more important when dealing with new materials.

Concurrently, an unceasingly growing interest in concepts such as Big-Data, Machine Learning or Data-Analytics has been experienced. Indeed, this interest is intrinsically motivated by an exhaustive development in both data-acquisition and data-storage systems. For instance, an aircraft may produce over 500 GB of data during a single flight. This panorama brings a perfect opportunity to the so-called Dynamic Data Driven Application Systems (DDDAS), whose main objective is to merge classical simulation algorithms with data coming from experimental measures in a dynamic way. Within this scenario, data and simulations would no longer be uncoupled but rather a symbiosis that is to be exploited would achieve milestones which were inconceivable until these days. Indeed, data will no longer be understood as a static calibration of a given constitutive model but rather the model will be corrected dynamicly as soon as experimental data and simulations tend to diverge. Several numerical algorithms will be presented throughout this manuscript whose main objective is to strengthen the link between data and computational mechanics. The first part of the thesis is mainly focused on parameter identification, data-driven and data completion techniques. The second part is focused on Model Order Reduction (MOR) techniques, since they constitute a fundamental ally to achieve real time constraints arising from DDDAS framework.

Résumé

Les sciences de l'ingénieur basées sur la simulation (*Simulation Based Engineering Science*, SBES) ont apporté des améliorations majeures dans l'optimisation, le contrôle et l'analyse inverse, menant toutes à une meilleure compréhension de nombreux processus se produisant dans le monde réel. Ces percées notables sont présentes dans une grande variété de secteurs tels que l'aéronautique ou l'automobile, les télécommunications mobiles ou la santé, entre autres. Néanmoins, les SBES sont actuellement confrontées à plusieurs difficultés pour fournir des résultats précis dans des problèmes industriels complexes. Outre les coûts de calcul élevés associés aux applications industrielles, les erreurs introduites par la modélisation constitutive deviennent de plus en plus importantes lorsqu'il s'agit de nouveaux matériaux.

Parallèlement, un intérêt sans cesse croissant pour des concepts tels que les données massives (*big data*), l'apprentissage machine ou l'analyse de données a été constaté. En effet, cet intérêt est intrinsèquement motivé par un développement exhaustif des systèmes d'acquisition et de stockage de données. Par exemple, un avion peut produire plus de 500 Go de données au cours d'un seul vol. Ce panorama apporte une opportunité parfaite aux systèmes d'application dynamiques pilotés par les données (*Dynamic Data Driven Application Systems*, DDDAS), dont l'objectif principal est de fusionner de manière dynamique des algorithmes de simulation classiques avec des données provenant de mesures expérimentales. Dans ce scénario, les données et les simulations ne seraient plus découplées, mais une symbiose à exploiter permettrait d'envisager des situations jusqu'alors inconcevables. En effet, les données ne seront plus comprises comme un étalonnage statique d'un modèle constitutif donné mais plutôt comme une correction dynamique du modèle dès que les données expérimentales et les simulations auront tendance à diverger. Plusieurs algorithmes numériques seront présentés tout au long de ce manuscrit dont l'objectif principal est de renforcer le lien entre les données et la mécanique computationnelle. La première partie de la thèse est principalement axée sur l'identification des paramètres, les techniques d'analyse des données et les techniques de complétion de données. La deuxième partie est axée sur les techniques de réduction de modèle (MOR), car elles constituent un allié fondamental pour satisfaire les contraintes temps réel découlant du cadre DDDAS.

Résumé étendu

La modélisation mathématique, les algorithmes de calcul et le calcul haute performance constituent une discipline connue sous le nom de science de l'ingénierie basée sur la simulation (*Simulation Based Engineering Science*, SBES). Celle-ci a amené le domaine de la simulation numérique dans une nouvelle ère où des améliorations majeures ont été réalisées dans l'optimisation, la modélisation multi-échelles, le contrôle et l'analyse inverse. De nombreuses techniques numériques différentes ont été développées tout au long de l'histoire pour améliorer la fiabilité des simulations numériques. Il convient de mentionner les techniques comme la méthode des éléments finis, la méthode des différences finies ou la méthode des volumes finis parmi de nombreuses autres approches, constituant trois des principales méthodologies qui sont encore largement utilisées de nos jours. Bien qu'ils diffèrent dans de nombreux aspects techniques, leur objectif commun est de transformer un ensemble donné d'équations aux dérivées partielles en un système discret d'équations algébriques qui est parfaitement compris par un ordinateur.

Si on analyse plus en détail comment un processus physique est modélisé, une classification entre deux ensembles différents d'équations peut être faite. Le premier ensemble contient les lois d'équilibre ou de conservation qui tentent de préserver ou d'équilibrer une quantité donnée dans l'ensemble du domaine comme c'est le cas, par exemple, de l'équilibre linéaire et angulaire ou de la conservation de l'énergie. Le deuxième ensemble contient les équations dites constitutives reliant ces quantités équilibrées ou préservées à des variables cinématiques pour rendre la solution unique. Par exemple, la loi de Hooke en mécanique des solides ou la fonction de rendement plastique en théorie de la plasticité peuvent être considérées comme des exemples d'équations constitutives. La dernière série d'équations a été utilisée avant même l'invention du calcul différentiel. Notamment, les observations de Tycho Brahe sur les mouvements des planètes conduisent Johannes Kepler à formuler ses trois équations constitutives aujourd'hui connues comme les trois lois de Kepler. Des années plus tard, Newton généralisera les lois de Kepler avec la formulation de la loi universelle de la gravité. En effet, il est juste de penser que les équations constitutives ont joué un rôle fondamental dans la modélisation de la plupart des processus physiques.

Cependant, à mesure que la compréhension d'un processus physique donné s'approfondit, sa modélisation d'un point de vue mathématique devient plus complexe. Ce scénario peut être apprécié dans le cadre de la mécanique des solides où la théorie de l'élasticité linéaire évolue vers des situations plus complexes telles que la mécanique non linéaire des solides ou les comportements inélastiques. Illustrant ce panorama, même si la loi de Hooke est largement utilisée en théorie de l'élasticité linéaire, le catalogue d'équations constitutives s'élargit lorsqu'on considère les matériaux hyper-élastiques, comme le Neo-Hookean, Saint Venant-Kirchhoff, Ogden,

Mooney-Rivling, etc. Le même effet peut être observé lorsque l'on considère l'inélasticité, où de nombreux modèles de plasticité ou d'endommagement différents ont été formulés pour se rapprocher du comportement réel du matériau. Un autre scénario dont la modélisation devient assez compliquée est un fluide complexe, comme par exemple le cas des écoulements de mousses. La modélisation numérique des processus de formage impliquant l'écoulement des mousses nécessite la prise en compte des différentes échelles de problèmes. Ainsi, dans les applications industrielles, une approche macroscopique est appropriée, alors que les paramètres macroscopiques d'écoulement dépendent de la structure cellulaire : taille, forme, orientation des cellules. De plus, la forme et l'orientation des cellules sont induites par le flux. Une description entièrement microscopique reste utile pour comprendre le comportement de la mousse et les changements topologiques induits par l'allongement ou la déformation des cellules, mais d'un point de vue industriel, les simulations microscopiques restent difficiles pour des applications pratiques impliquant des flux dans des géométries 3D complexes. Dans la pratique, les modèles constitutifs macroscopics viscoélastiques, où la microstructure de la mousse est représentée à partir de descripteurs de microstructure appropriés, sont régis par la cinématique macroscopique de l'écoulement. La dérivation de ces comportements constitutifs est détaillée à l'annexe A, ce qui a certainement renforcé les soupçons quant à la complexité liée à la modélisation de nouveaux matériaux.

Il est donc raisonnable de penser que la nature phénoménologique des équations constitutives peut introduire des erreurs de modélisation, ce qui entraîne des inadéquations entre les simulations et les résultats expérimentaux. De plus, cette erreur de modélisation tend à augmenter naturellement dès que la complexité du système augmente. Par conséquent, beaucoup d'efforts ont été consacrés à réduire toutes les sources d'erreur dans les simulations numériques, non seulement l'erreur de modélisation, mais aussi les erreurs de discrétisation. En effet, un scénario idéal où les expériences et les simulations numériques seraient en parfait accord permettrait d'aller au-delà des limites de conception actuelles. Non seulement parce que l'exploration de nouveaux prototypes serait quantitativement moins coûteuse, mais aussi parce que les connaissances à l'intérieur de la simulation numérique sont beaucoup plus riches que les données recueillies par les capteurs expérimentaux. Malheureusement, ce monde idyllique n'existe pas de nos jours, donc les outils numériques doivent être adaptés en conséquence pour faire face à cette problématique. Cette transformation des simulations numériques standard est revendiquée car, traditionnellement, les simulations sont principalement statiques dans le sens où les équations constitutives sont calibrées avant de faire la simulation sans pouvoir les modifier dynamiquement dès que les données expérimentales ont tendance à différer des simulations. Aujourd'hui, la situation est radicalement différente, les données sont beaucoup plus abondantes (et précises) que les modèles existants, et un nouveau paradigme émerge dans les sciences de l'ingénieur et la technologie, les systèmes d'application dynamiques pilotés par les données (*Dynamic Data Driven Application Systems*, DDDAS). Par exemple, les expériences de physique des hautes énergies produisent environ 1Pb de données par jour, alors qu'en 2012, 162 000 articles ont été publiés dans des re-

vues de science des matériaux. Ces faits indiquent clairement que les données sont de plus en plus abondantes alors que le catalogue des matériaux possibles s'élargit énormément. Ainsi, une fusion cohérente et efficace entre les données et les outils de simulation standard constitue la voie d'investigation actuelle pour de nombreux centres de recherche. Cette possibilité sans précédent de déterminer ou d'enrichir directement les connaissances à partir de données ou, en d'autres termes, d'extraire des modèles d'expériences de façon automatisée, est suivie avec grand intérêt dans de nombreux domaines des sciences et de l' ingénierie. Bien que presque classiques dans d'autres domaines scientifiques comme l'économie, la sociologie, de données massives sont arrivées avec un retard important dans le domaine de la mécanique computationnelle. Il convient de noter que dans notre domaine la quantité de données disponibles n'est très souvent pas aussi importante, et nous parlons donc de techniques axées sur les données plutôt que de techniques axées sur les données massives.

D'autre part, les outils liés aux techniques avancées de classification deviennent cruciaux dans de nombreux domaines où les modèles, les bases d'approximation ou les paramètres sont ajustés en fonction de l'espace latent, qui est un espace réduit représentant la physique fondamentale du système. Ils permettent de définir une modélisation hiérarchique, orientée vers un but et, en fin de compte, de définir un nouvel ensemble de coordonnées réduites portant la plus grande partie de l'information. Les techniques d'apprentissage machine sont capables d'extraire la structure multiple dans laquelle vit la solution de problèmes d'ingénierie complexes et couplés. Ainsi, des paramètres non corrélés peuvent être efficacement extraits des données collectées, provenant soit de simulations numériques, soit d'expériences. Dès que des paramètres non corrélés sont identifiés (constituant le niveau d'information), la solution du problème peut être prévue à de nouveaux emplacements de l'espace paramétrique, en utilisant des schémas d'interpolation adéquats. Dans un cadre différent, des solutions paramétriques peuvent être obtenues dans un cadre adéquat capable de contourner la malédiction de la dimensionnalité pour toute valeur des paramètres non corrélés du modèle. Il est important de noter à ce stade que l'interpolation est une question délicate lorsque la solution définit un variété (*manifold*) incluse dans l'espace dimensionnel élevé. Dans ce cas, pour définir une interpolation précise, il faut procéder par interpolation sur la variété. L'apprentissage sur la variété a également été utilisé pour la construction de modèles réduits, pour y définir des solutions paramétriques, ou simplement pour définir avec succès des applications de mécanique computationnelle pilotées par les données. Dans ces derniers cas, les équations constitutives traditionnelles ont été remplacées par un ensemble de données collectées.

De même, la réduction de modèles (MOR) revêt une importance capitale pour l'ingénierie basée sur la simulation. Ces techniques permettent de résoudre efficacement des modèles mathématiques complexes, grâce à l'utilisation de bases d'approximations adaptées pour décrire leurs solutions. Parmi les nombreuses techniques MOR existantes, la Proper Orthogonal Decomposition (POD), la Proper Generalized Decomposition (PGD) et Reduced Basis (RB) sont largement considérées dans

une variété d'applications. En effet, les techniques MOR pourraient être considérées comme un allié naturel pour explorer de nouvelles voies où les données, les modèles et les simulations constituent un cadre solide pour stimuler les percées technologiques.

A la lumière de cette révolution des données, il est facile de comprendre les raisons pour lesquelles le concept d'Industrie 4.0 suscite beaucoup d'intérêt ces derniers temps. La possibilité de créer des processus intelligents où les données jouent un rôle majeur afin d'améliorer l'efficacité est une option plutôt intéressante. Cette amélioration serait obtenue grâce à la fusion adéquate des données collectées et du processus à optimiser. Ces données collectées peuvent être, par exemple, des variables quantitatives telles que l'évolution historique des températures, de la pression ou des déplacements ou qualitatives comme les fabricants, les matériaux, le nom de l'opérateur, etc. Cet ensemble de données alimentera le processus lui-même en déterminant les actions à entreprendre en temps réel. Cependant, comme cette Industrie 4.0 est encore une idée émergente, plusieurs aspects doivent être traités et améliorés avant de la transformer en réalité. A première vue, ce concept d'usine intelligente peut sembler très dépendant du problème, ou en d'autres termes, chaque problème peut dépendre en grande partie du processus à optimiser. Cependant, les principaux inconvénients associés au travail avec les données pourraient être classés comme suit :

- *Rationalisation des données.* La raison d'être des données massives favorise la collecte d'un ensemble de données aussi vaste que possible. Cependant, cet excès de données peut être contre-productif lorsqu'on essaie de dévoiler des relations cachées, car l'extraction de ces relations sous-jacentes sera beaucoup plus coûteuse. Par conséquent, il est important de toujours garder à l'esprit que les données massives devraient évoluer vers des données intelligentes, où des questions telles que pourquoi - quand - où les données seront-elles recueillies sont traitées avant d'être réellement recueillies. La formulation de ce type de questions peut déboucher sur une simplification considérable du problème en sélectionnant les variables d'entrée et de sortie les plus importantes qui contrôlent le processus.

- *Données réduites.* Malgré le fait que les grandes données impliquent intrinsèquement de travailler avec une énorme quantité de données. La vérité est que parfois la quantité de données accessibles n'est pas aussi riche que nécessaire. Cette dégénérescence vers la limite inférieure des données peut être causée par plusieurs raisons. Par exemple, un phénomène *d'espace vide* peut apparaitre lorsqu'on considère des espaces dimensionnels élevés, ce qui se produit lorsque l'on considère toutes les variables affectant le processus. Un autre cas de données réduites peut apparaître lorsque l'on considère les données fournies par les capteurs puisqu'elles sont placées à des endroits spécifiques du processus. Par conséquent, les variables d'état ne seront pas définies partout dans le domaine mais seulement à certains emplacements de capteurs.

- *Algorithmes numériques rapides.* De nouveaux algorithmes destinés à combi-

ner les processus de données et de simulations standard sont intégrés dans les systèmes d'application dynamiques pilotés par les données (DDDAS). En effet, le mot dynamique incite à faire ces corrections de données en temps réel, ou au moins aussi dynamique que possible. Il semble naturel d'utiliser les méthodologies MOR pour accélérer les algorithmes autant que possible. Ils peuvent être utilisés comme un outil efficace pour explorer les espaces à haute dimension, par exemple, lorsqu'il s'agit d'algorithmes d'identification de paramètres.

- *Fusion de données cohérentes.* L'utilisation des données est pratique, malheureusement, il y a des situations dans lesquelles les données disponibles ne remplissent pas certaines conditions *per se*. Par exemple, nous devrions être réticents ou au moins conscients de situations telles qu'une identification de conductivité non positive à partir d'un profil de flux température-chaleur tout comme un tenseur d'élasticité défini non positif construit à partir d'un ensemble de données de force de déplacement. Ces situations dégénérées peuvent conduire à des perspectives où les données ne sont plus un substitut naturel des modèles existants, mais plutôt un allié pour enrichir les modèles existants en préservant la cohérence thermodynamique.

Les contributions présentées dans cette thèse abordent certaines difficultés présentées dans la liste ci-dessus, en développant des algorithmes numériques qui renforcent le lien entre les données et les sciences de l'ingénieur basées sur la simulation. Plus de détails sur chaque contribution sont précisés dans la section suivante.

Structure de la thèse

Cette thèse est organisée en deux parties distinctes, abordant les problèmes présentés précédemment. En bref, la partie I constitue un premier pas vers une fusion efficace et cohérente des domaines de la science des données et de la mécanique computationnelle. La partie II se concentre sur les techniques MOR, car elles constituent un allié fondamental pour atteindre les contraintes de temps réel découlant du cadre DDDAS. La structure générale de la thèse est donnée par :

- ***Partie I :*** *Premiers pas vers une fusion efficace des données et de la mécanique computationnelle*

    - ***Chapitre 2 :*** *Techniques axées sur les données*
    - ***Chapitre 3 :*** *Techniques de complétion des données*

- ***Partie II :*** *Avancées sur les techniques de réduction de modèles*

    - ***Chapitre 4 :*** *Multi PGD basé sur la méthode de partitionnement d'unité*
    - ***Chapitre 5 :*** *Applications des séparations plan/hors-plan PGD*

Chapitre 2 : Techniques axées sur les données

Ce chapitre traite de l'utilisation et de l'identification des relations constitutives. La relation constitutive n'est plus comprise comme une équation dans son format fermé, mais comme un nuage de points ou un ensemble de données. Une première tentative de fusionner cette base de données constitutive avec un schéma numérique a été étudiée dans le cadre d'élasticité. Dans ce cas, le solide est traité comme un ensemble de barres 1D qui sont calibrées avec des courbes de contrainte/déformation standard. Ces courbes de contrainte/déformation portent toutes les informations matérielles constitutives. Ensuite, un schéma itératif entre ces courbes expérimentales et l'équilibre de quantité de mouvement est résolu. Certes, la simulation guidée par les données consiste à utiliser directement des données pour effectuer des simulations numériques. Ces simulations utiliseront des lois universelles tout en minimisant le besoin de modèles explicites, souvent phénoménologiques. Cette approche est particulièrement intéressante lorsqu'il s'agit de matériaux d'ingénierie complexes (métamatériaux), pour lesquels les relations constitutives deviennent difficiles à écrire, car il existe (trop) de possibilités, et la nature intime de la plupart d'entre eux reste inaccessible et/ou confidentielle.

Même si, d'un point de vue conceptuel, un ensemble de points représentant le comportement constitutif ou l'équation réelle sous sa forme fermée portent la même information, sa transformation en base de données permet d'extraire les éléments les plus pertinents de la collecte de données. Les techniques d'apprentissage machine, et plus spécifiquement la réduction de dimensionnalité non linéaire, comme par exemple le *Locally Linear Embedding* (LLE), la kernel-PCA (l'équivalent non linéaire de l'analyse en composantes principales —PCA), appelée k-PCA, la local-PCA, et d'autres, nous permettent de supprimer les corrélations dans les données. Ces données, libres de toute corrélation, constituent l'information réelle, souvent très limitée par rapport aux grandes quantitées de données dont elles ont été extraites. Dans de nombreux modèles, l'extraction de paramètres non corrélés demeure une question délicate. C'est le cas des paramètres décrivant des microstructures ou des formes par exemple, souvent appelés paramètres latents. Dès que les paramètres non corrélés sont extraits, deux options principales ont été envisagées à ce jour : (i) lorsqu'un nouveau cas, non inclus dans les données, doit être analysé, sa solution est simplement interpolée sur la varieté (construit à partir des données de formation) de ses plus proches voisins afin que les décisions puissent être prises en temps réel ; et (ii) une solution paramétrique explicite peut être construite en utilisant les paramètres non corrélés juste extraits afin de pouvoir être détaillés en temps réel.

La section 2.1 traite de l'utilisation de l'ensemble des données constitutives du cadre de l'élasticité. Le solide n'est plus compris comme un ensemble de barres 1D mais plutôt comme un solide continu où les contraintes et les déformations, deux tenseurs de deuxième ordre, sont reliées par un tenseur de quatrième ordre appelé le module élastique. En conséquence, l'ensemble de données constitutives a douze coordonnées en 3D, six pour définir la contrainte et six autres pour définir les états de déformation en supposant la symétrie, où de multiples techniques d'apprentissage

peuvent être appliquées. Ensuite, plusieurs techniques sont présentées pour fusionner l'ensemble de données constitutives avec la conservation de la quantité de mouvement afin d'être cohérent avec les algorithmes standard d'éléments finis. La présente section n'aborde pas toutes les difficultés techniques liées à la production de données ou à l'obtention d'expériences adéquates. Au contraire, nous développons une méthode dans laquelle ce flux de données joue le rôle d'une équation constitutive, sans avoir besoin d'une adaptation phénoménologique à un modèle fermé prescrit.

La section 2.2 traite de l'un des goulets d'étranglement actuels des simulations pilotées par les données : l'énorme quantité de données requises, dont certaines sont inaccessibles depuis les installations de test actuelles. En effet, il est difficile d'explorer l'ensemble de l'espace contrainte/déformation au moyen d'expériences homogènes où seuls des états de contrainte limités sont accessibles, c'est-à-dire des charges uniaxiales ou biaxiales. Cette difficulté peut être contournée dans de nombreux cas, et au moins atténuée dans les autres cas, en considérant des tests complexes (ou hétérogènes), où coexistent à l'intérieur du matériau de nombreux états de contrainte/déformation différents. Ensuite, une approche inverse guidée par les données permet d'extraire une grande quantité de données des expériences complexes. Une alliance subtile de machines de test, de dispositifs de collecte de données et d'ordinateurs puissants pour traiter ces énormes quantités de données de diverses manières (machine et apprentissage approfondi) permet de dévoiler la multiplicité constitutive de l'expérience.

La section 2.3 traite d'une autre application dans le cadre de la justification fondée sur les données. En effet, la mise à l'échelle des comportements de microstructures hétérogènes pour définir des milieux macroscopiques efficaces est d'un intérêt majeur dans de nombreux domaines de la mécanique computationnelle, en particulier ceux liés au génie des matériaux et des procédés. Dans ce travail, nous explorons la possibilité de définir un comportement macroscopique à partir d'un ensemble de données basé sur des calculs microscopiques, puis de l'utiliser directement pour effectuer efficacement des simulations à l'échelle macroscopique. Nous considérons dans ce travail la mise à l'échelle des écoulements non newtoniens en milieu poreux, et plus particulièrement ceux impliquant des suspensions de fibres courtes. D'autre part, lorsqu'on considère les écoulements de fluides simples ou complexes en milieu poreux, la situation devient difficile car la physique rencontrée à l'échelle micro diffère de celle que l'on suppose à l'échelle macro. Ainsi, l'écoulement micro-échelle est régi par un problème de Stokes, qui devient non linéaire dès que la viscosité du fluide dépend de la vitesse de déformation. Dans le cas de suspensions de tiges (fibres, microfibres, nanofibres ou nanotubes), l'écoulement à petite échelle est régi par un problème de Stokes anisotrope où la viscosité est localement très anisotrope. Il est toutefois inutile d'homogénéiser les paramètres d'écoulement microscopiques afin de définir une viscosité efficace à l'échelle macroscopique. Ainsi, la voie la plus valable consiste à considérer autant que possible des modèles visqueux plus ou moins complexes pour capturer les principales caractéristiques du fluide et de l'écoulement à l'échelle microscopique, et d'en obtenir une perméabilité de type Darcy disponible à l'échelle macroscopique pour les fournir en entrée aux logiciels

de simulation classiques.

Chapitre 3 : Techniques de complétion des données

Comme cela a été détaillé dans la section 2.2, un problème récurrent des applications pilotées par les données est la génération de données, celles-ci pouvant être inaccessibles ou coûteuses à obtenir. Par conséquent, le développement d'algorithmes de complétion de données où les surfaces de réponse sont créées à partir d'un ensemble de données existant est d'une importance vitale. Le problème avec les systèmes de grande dimension est que les données dans ces systèmes sont souvent rares (en raison précisément de la nature de grande dimension de l'espace de phase) alors que le système a très souvent, au contraire, de faibles caractéristiques dimensionnelles.

Les techniques MOR ont amélioré les stratégies traditionnelles basées sur le DoE (*Design of Experiments*), qui ont permis de définir des métamodèles, des modèles de substitution ou des surfaces de réponse. Dans ces cas, des expériences ou des solutions de calcul coûteuses sont réalisées pour un échantillonnage des états possibles du système, à partir duquel un modèle simplifié reliant les entrées aux sorties est élaboré. Les principales difficultés associées à cette procédure concernent la meilleure stratégie d'échantillonnage et le schéma d'interpolation le plus adéquat pour faire des prédictions partout dans l'espace de conception à partir de la seule connaissance des quelques scénarios analysés. L'hypercube latin et le Krigeage sont deux réponses habituelles à ces questions. Cependant, d'autres questions restent ouvertes, telles que la vérification des modèles (estimation et délimitation des erreurs) ainsi que la définition de stratégies adaptatives permettant de réduire ces erreurs localement ou globalement. Même s'il existe une panoplie de propositions et de stratégies appliquées, la plupart d'entre elles dépendent des problèmes et ne sont pas robustes et fiables. Comme nous venons de le mentionner, les méthodes de réduction de modèle ont établi des voies pour atteindre des objectifs similaires tout en contournant les principaux problèmes qui viennent d'être mentionnés, pour finalement définir un "abaque", construit hors ligne et utilisé efficacement en ligne pour une conception robuste.

Même si, comme nous venons de l'indiquer, les technologies MOR rendent possible de meilleures approches, leur principal inconvénient est qu'elles restent souvent trop intrusives. Aujourd'hui, les travaux les plus récents concernant les techniques MOR se concentrent sur les algorithmes non-intrusifs. Cependant, d'un point de vue pragmatique, toutes ces propositions restent moins directes que les méthodologies habituelles du DoE, la dernière consistant simplement à évaluer le modèle à différents points de l'espace de conception en utilisant des solveurs commerciaux standard (adaptés au problème en question) et ensuite simplement à interpoler ces solutions à tout autre point. Malgré les difficultés conceptuelles qui viennent d'être évoquées, la procédure est très simple et a attiré la faveur des ingénieurs, concepteurs et praticiens.

Ainsi, le tableau d'ensemble pourrait être formulé comme suit : l'échantillonnage direct pourrait-il conduire à une solution paramétrique robuste et fiable ?

La réponse à cette question a traditionnellement été abordée de diverses manières. Tout d'abord, les méthodologies basées sur la surface de réponse proposent une sorte de procédure adaptative lorsque le modèle est affiné par zoom lorsque la solution s'approche de la solution optimale par rapport à un couple modèle donné / critère d'optimisation. Toutefois, un tel approvisionnement nécessite un calcul en ligne car la représentation fine n'est pas *a priori* disponible et doit être construite en ligne pendant le processus d'optimisation.

Cela dit, une technique efficace de complétion des données dans un cadre axé sur les données devrait répondre aux exigences suivantes :

- *Ensemble de données non structuré* : cette caractéristique confère à la méthode une grande polyvalence. En effet, lorsque l'évaluation de la surface de réponse exige beaucoup d'efforts de calcul, le recyclage des évaluations précédentes de la surface de réponse, qui ne coïncident pas avec une structure particulière des données, peut être très utile. De plus, disposer d'algorithmes capables de travailler avec des ensembles de données non structurés est d'une importance vitale pour certaines applications où les emplacements paramétriques sont difficiles à explorer. Par exemple, il est plus facile d'explorer les états de charge uniaxiale dans la variété constitutive que les états biaxiaux, et même plus facile que les états triaxiaux.

- *Robustesse de haute dimensionnalité* : la plupart des techniques de complétion de données, comme celles basées sur la triangularisation, souffrent lorsqu'il s'agit de données multidimensionnelles simplement parce qu'un maillage de grande dimension doit être généré. Néanmoins, la séparation des variables pourrait être une technique attrayante pour contourner, ou à tout le moins atténuer, le problème des espaces à haute dimension.

- *Quantité limitée de données* : la robustesse des techniques de complétion de données lorsque la quantité de données est limitée est cruciale. En effet, les situations dans lesquelles le coût d'agrandissement de l'ensemble de données est inabordable exigeront une technique numérique capable de fournir une surface de réponse raisonnable sur la base de l'ensemble de données épars donné.

La section 3.1 présente une nouvelle technique de complétion de données appelée sparse Proper Generalized Decomposition (sPGD). Il combine la logique de la Proper Generalized Decomposition (PGD), où la séparation des variables est utilisée pour contourner ou atténuer la malédiction de la dimensionnalité avec une procédure de colocalisation coïncidant avec l'ensemble de données non structuré. Ce faisant, aucune information n'est requise à part de l'ensemble de données, ce qui permet de construire une surface de réponse efficace basée sur les données disponibles.

La section 3.2 décrit trois applications différentes d'intérêt industriel dans lesquelles la technique sPGD expliquée dans la section 3.1 a été appliquée. La première application en collaboration avec *Gestamp* concerne l'industrie automobile, où la

forme géométrique d'un composant à l'intérieur du véhicule est optimisée pour garantir la sécurité des passagers. La seconde application en collaboration avec *ESI Group* vise à corriger les modèles de plasticité existants au moyen de quantités observables, c'est-à-dire des mesures synthétiques de déformation et de déplacement. La troisième application relie les paramètres géométriques définissant la surface d'un matériau composite à sa performance dans le procédé de pose automatisée de rubans (*Automated Tape Placement* ATP). En effet, les paramètres géométriques tels que la courbure ou la rugosité sont extraits de matériaux composites réels sous la forme d'un profilomètre. Plusieurs techniques de clustering et de complétion de données, dont le sPGD, sont utilisées pour relier les premiers paramètres à la performance du matériau composite.

La section 3.3 étend le concept de techniques d'identification sparses, déjà introduit pour identifier le modèle caché dans un ensemble de données, à d'autres applications dans le cadre de la mécanique computationnelle. Ce concept a été largement appliqué dans d'autres domaines comme la compression d'images, en évoquant le concept de caméra à pixel unique. Cette caméra à pixel unique ou à détection comprimée prétend être capable de reconstruire une image donnée simplement en connaissant l'information à quelques pixels. Le même raisonnement est appliqué pour reconstruire une surface de réponse paramétrique tout comme la possibilité d'appliquer la méthodologie à des problèmes évolutifs dans le temps.

Chapitre 4 : Multi PGD basé sur la méthode de partitionnement d'unité

Les techniques de réduction de modèles (MOR) deviennent très utiles dans le cadre d'un système d'application basé sur des données dynamiques où la prise de décision doit se faire de manière dynamique. Ces dernières années, les techniques MOR ont montré qu'un nombre minimum de degrés de liberté soigneusement choisis sont suffisants pour une solution précise des EDP, réduisant ainsi le coût des discrétisations FEM standard. Au lieu de choisir des bases polynomiales génériques, les techniques MOR construisent des fonctions de base *ad hoc* suivant différentes techniques. Par exemple, la Proper Orthogonal Decomposition (POD) construit une base plus adéquate à partir d'un ensemble de snapshots pré-calculés de la solution PDE du modèle complet,

$$u(x,t) = \sum_{i=1}^{N} \alpha_i(t)\phi_i(x), \tag{1}$$

où, comme dans le contexte des éléments finis, α_i est un ensemble de coefficients dépendants du temps, et $\phi_i(x)$ sont des fonctions de base indépendantes du temps obtenues par un traitement statistique des snapshots du système. Par exemple, de nombreuses techniques MOR utilisent les fonctions propres les plus énergiques de la matrice d'autocorrélation des snapshots pour construire ces $\phi_i(x)$. Celles-ci jouent un rôle similaire à celui des fonctions de forme des éléments finis, bien qu'elles soient globales au lieu d'être locales. D'autres techniques, telles que les méthodes à base

réduite, utilisent quelques snapshots de la solution d'ordre complet comme base pour la solution approximative du système. Ces snapshots sont calculés de manière gourmande, à des instants (ou paramètres) où l'erreur dans l'approximation est maximale.

Il est important de noter que Eq. 1 constitue en fait une expression séparée ou affine de la solution—notez la séparation espace-temps, qui tient également dans les approximations par éléments finis—. Cette méthodologie s'est avérée très efficace pour une grande variété de problèmes de grande dimension allant de la résolution de l'équation de Fokker-Planck aux réponses hépatiques spécifiques au patient, en passant par la dynamique structurelle, la rhéologie computationnelle ou, plus généralement, tout problème paramétrique qui pourrait être écrit sous une forme separé. Il y a cependant des situations où la solution est très inséparable. En d'autres termes, la solution du problème réside dans une varieté non linéaire. Dans cette situation, la technique MOR consiste à projeter la solution sur l'espace tangent à la varieté en un point donné (temps ou paramètre). Ceci conduit à de mauvaises propriétés d'approximation loin du point tangent, à moins que des techniques spéciales ne soient choisies.

Une façon d'atténuer ce problème est de suivre la même philosophie que les méthodes de réduction de dimensionnalité non linéaire. Par exemple, la *Locally Linear Embedding* (LLE) tente de dévoiler les variables latentes en imposant une variation linéaire locale à la fonction, qui changera de voisinage en voisinage. Une autre technique qui traite des problèmes non linéaires est la *kernel-PCA* (k-PCA). Dans ce cas particulier, la définition d'une fonction noyau efficace permet de projeter les snapshots dans des espaces de grande dimension (potentiellement infinie) dans lesquels la varieté de solution est plate. Dans ce cas particulier, les techniques d'interpolation standard fonctionnent bien. Lors de l'étude du cas d'une source en mouvement dans un problème de transfert thermique transitoire, le problème de l'inséparabilité de la solution a été contourné par la réalisation d'une partition du domaine temporel, en dédiant un DPI différent pour chaque partition. Cependant, l'imposition des conditions d'interface deviendra une tâche fastidieuse lorsqu'il s'agit de partitions impliquant des variables autres que le temps.

La section 4.1 propose une méthodologie où différents PGD sont combinés grâce à l'introduction de fonctions de fonctions de forme FEM macro satisfaisant la partition de l'unité. Ces fonctions de forme macro, à savoir les fonctions polynomiales par morceaux FEM, sont chargées de contrôler le chevauchement entre les différents PGD, sans avoir besoin de spécifier les conditions d'interface entre eux. De plus, le fait que la solution devienne localement plus linéaire (aussi plus séparable) facilite la tâche de chacun des PGD, réduisant drastiquement le nombre de modes pour obtenir une solution précise.

La section 4.2 étend la méthodologie proposée dans la section 4.1 pour traiter les problèmes multi-échelles efficacement. De nombreux problèmes en mécanique computationnelle présentent un comportement multi-échelles où les effets à la microéchelle influencent ceux à la macro-échelle et vice-versa. Par conséquent, le traitement des différentes échelles du problème devient très important pour parvenir à

une solution précise. Un tel comportement multi-échelle apparaît, par exemple, dans le domaine spatial lorsqu'il s'agit de métamatériaux dont la structure est définie au niveau micro. Plusieurs techniques ont été appliquées tout au long de l'histoire comme l'homogénéisation définissant le modèle constitutif à la micro-échelle en fonction d'un volume représentatif qui satisfait au principe de Hill-Mandel. Cette méthode s'est avérée très efficace pour contourner le coût de calcul prohibitif de méthodes comme FEM^2. Cependant, une hypothèse forte sous-tend cette technique d'homogénéisation, qui exige une séparation claire des échelles entre les effets macro et micro.

En effet, la section 4.2 propose une méthodologie dans laquelle des fonctions de forme macro qui respectent la partition d'unité ont permis à la méthode de combiner différents PGDs de manière cohérente. De plus, le chevauchement entre les PGDs résout automatiquement la transition entre les différentes échelles.

Chapitre 5 : Applications des séparations plan-hors-plan PGD

De nombreux systèmes mécaniques et structures complexes impliquent des pièces en plaques et coques dont la principale particularité est d'avoir une dimension caractéristique (celle liée à l'épaisseur) beaucoup plus petite que les autres (dimensions en plan). L'introduction d'hypothèses cinématiques et mécaniques appropriées permet de réduire le problème mécanique général 3D à un problème 2D impliquant les coordonnées en plan. C'était la voie utilisée pour dériver les théories des poutres, des plaques et des coques en mécanique des solides, qui ont ensuite été étendues à beaucoup d'autres domaines de la physique, comme les écoulements dans des espaces étroits, les problèmes thermiques ou électromagnétiques des stratifiés, parmi tant d'autres. Pourtant, lorsque la physique est appliquée à des domaines dégénérés, comme les plaques ou les coques, et qu'aucune hypothèse de simplification acceptable n'est disponible pour réduire leur complexité à la 2D, des solutions entièrement 3D semblent obligatoires. C'est le cas, par exemple, lorsque l'on considère l'endommagement dynamique progressif des stratifiés composites où une description riche à travers l'épaisseur pourrait être extrêmement utile.

Lorsque ces modèles 3D deviennent obligatoires, l'approximation des différents champs implique des milliers de nœuds répartis dans le sens de l'épaisseur, et par conséquent des millions de nœuds pour représenter les champs dans la pièce. Aujourd'hui, la solution de ces modèles 3D fins reste insoluble malgré les progrès impressionnants réalisés en modélisation, analyse numérique, techniques de discrétisation et informatique au cours des dernières décennies. Les techniques de discrétisation standard basées sur des maillage échouent au vu du nombre excessif de degrés de liberté impliqués dans la discrétisation. En effet, des mailles très fines sont nécessaires dans le sens de l'épaisseur (malgré sa dimension réduite) et souvent aussi dans le sens du plan pour éviter soit des mailles trop déformées, soit parce que certains procédés (par exemple les micro-ondes) nécessitent des représentations fines dans le plan.

Afin d'alléger la complexité des calculs associés, les auteurs ont proposé il y

a quelques années de séparer l'espace physique. Ainsi, une décomposition dans le plan/hors-plan a été proposée pour résoudre dans les plaques les flux 3D se produisant dans les procédés RTM (moulage par transfert de résine), puis pour résoudre les problèmes d'élasticité des plaques et des coques. Dans ces cas, la solution 3D complète a été obtenue à partir de la solution d'une séquence de problèmes 2D (ceux impliquant les coordonnées dans le plan) et 1D (ceux impliquant les coordonnées relatives à l'épaisseur de la plaque). Il est important de souligner le fait que ces approches sont radicalement différentes des approches standard. Nous proposons un solveur 3D capable de calculer les différents champs 3D inconnus sans qu'il soit nécessaire d'introduire aucune hypothèse. L'avantage le plus remarquable est que les solutions 3D peuvent être obtenues avec un coût de calcul caractéristique des solutions 2D standard.

Ce chapitre a pour but de présenter quelques nouvelles applications de la représentation séparée dans le plan/hors-plan.

La section 5.1 vise à séparer les dimensions spatiales dans le cadre de l'électromagnétique. La principale motivation de cette étude numérique réside dans l'énergie et le temps qu'exige un autoclave. De nombreux procédés de fabrication de composites impliquent l'utilisation de l'autoclave pour assurer, après le formage, l'application simultanée de la température et de la pression pour obtenir à la fois une consolidation et une faible porosité. Le coût élevé de ces procédés en termes d'énergie et de temps a récemment conduit à la recherche d'autres technologies pour remplacer les autoclaves. Parmi celles-ci, le chauffage par micro-ondes est considéré comme un bon candidat pour accélérer les processus de fabrication. Ici, en fonction des propriétés électromagnétiques du renforcement, deux sources thermiques peuvent coexister : les pertes diélectriques et l'induction, cette dernière étant liée aux renforts électriquement conducteurs (par exemple les fibres de carbone). Dans le contexte de l'électromagnétisme, la situation était similaire à celle que l'on rencontre en mécanique structurelle, où l'on a discuté de la section 2D, des solveurs planaires 2,5D et des solveurs arbitraires 3D. Les simulateurs de guides d'ondes utilisent largement des formulations 2.5D dont l'une des dimensions a été éliminée en supposant une évolution particulière du champ électromagnétique dans cette direction. Les formulations 2.5D ont également été largement prises en compte pour l'analyse des circuits imprimés. Des modèles microscopiques, mésoscopiques et macroscopiques ont été proposés et largement pris en compte dans les applications d'ingénierie, en particulier dans celles impliquant des stratifiés. De nombreux travaux ont porté sur l'homogénéisation microscopique des systèmes électromagnétiques, en particulier des matériaux composites. A l'échelle mésoscopique, les stratifiés ont été abordés en vue de définir des approches macroscopiques. Trois voies principales ont été envisagées : (i) homogénéisation du stratifié ; (ii) éléments de coque monocouches et multicouches, où l'impédance de coque a été dérivée analytiquement en supposant un certain nombre d'hypothèses simplificatrices, puis couplée à la solution du problème électromagnétique dans le domaine externe afin d'obtenir les champs électromagnétiques sur les surfaces de coque, et en calculant à partir de celles-ci toute la physique interne (il est important de noter que les hypothèses supposées pourraient, dans certaines circons-

tances complexes, se révéler défectueuses) (iii) éléments spéciaux pour le maillage des stratifiés lors de l'adressage explicite de la solution à l'intérieur, cependant, pour représenter finement les couches limites à l'intérieur des couches, la complexité de calcul due à la résolution de maillage requise pourrait devenir excessive. C'est la raison pour laquelle la séparation des dimensions spatiales semble appropriée dans ce cas particulier.

La section 5.2 introduit une nouvelle représentation hybride explicite/implicite dans le plan à partir du plan pour les problèmes dynamiques définis dans des domaines en forme de plaques qui calculent efficacement la solution 3D et où les contraintes de stabilité sont exclusivement déterminées par les discrétisations plus grossières en plan. Il existe une vaste littérature sur la dynamique structurelle, couvrant différentes techniques de discrétisation et procédures d'intégration temporelle. Lorsqu'on considère une analyse implicite, la solution à chaque pas de temps nécessite quelques itérations pour faire respecter l'équilibre. Contrairement aux schémas implicites, les schémas explicites ne nécessitent pas d'itération car les accélérations nodales sont résolues directement, et à partir desquelles les vitesses et déplacements sont calculés par simple intégration. Le principal inconvénient des simulations explicites est que le pas de temps doit vérifier la condition de stabilité, qui diminue avec la taille de l'élément. Au contraire, les intégrations élastodynamiques implicites deviennent inconditionnellement stables, c'est-à-dire qu'il n'y a pas de limite dans le pas de temps à considérer en matière de stabilité. Ainsi, les pas de temps implicites sont généralement de plusieurs ordres de grandeur plus grands que ceux considérés dans les intégrations temporelles explicites. Cependant, l'intégration implicite nécessite la solution de systèmes linéaires plusieurs fois à chaque étape de chargement lorsqu'il s'agit de modèles non linéaires. D'autre part, comme les techniques explicites ne nécessitent pas l'inversion d'une matrice, elles s'adressent facilement aux non-linéarités (non-linéarités de contact ou non-linéarités matérielles) Un schéma hybride a été proposé qui considère le domaine composé de deux parties dans lesquelles des intégrations temporelles explicites et implicites sont applicables. Dans ce travail, nous introduisons une nouvelle représentation hybride explicite/implicite, explicite et implicite, dans le plan et hors du plan, des problèmes dynamiques définis dans des domaines en forme de plaques qui calculent efficacement les solutions 3D et où les contraintes de stabilité sont exclusivement déterminées par les discrétisations plus grossières dans le plan.

La section 5.3 se concentre sur les composites thermoplastiques puisqu'ils sont des matériaux structuraux privilégiés en raison de leurs excellentes propriétés de tolérance aux dommages, de cycles de fabrication plus courts et de la facilité de soudage. L'un des matériaux précurseurs pour fabriquer des pièces composites thermoplastiques est un préimprégné unidirectionnel (UD) qui consiste en fibres continues alignées préimprégnées de résine thermoplastique. A l'état fondu, le préimprégné UD peut être considéré comme une fibre inextensible entourée d'une matrice visqueuse incompressible, et peut donc être modélisé comme un fluide isotrope transversal. Ces stratifiés UD sont généralement empilés dans les orientations souhaitées pour créer un stratifié composite. En effet, une application de la représentation sépa-

rée dans le plan hors plan pour comprimer les flux de stratifiés à fibres continues est présentée. Plus en détail, l'équation constitutive du pli est modélisée comme un fluide transversalement isotrope, qui doit satisfaire à la fois l'inextensibilité de la fibre et l'incompressibilité du fluide. Lorsque le stratifié est comprimé, la cinématique d'écoulement présente une dépendance complexe le long de l'épaisseur du stratifié, ce qui nécessite une description détaillée de la vitesse à travers l'épaisseur. Dans des travaux précédents, la solution utilisant une représentation séparée dans le plan hors plan au sein du cadre PGD - Poper Generalized Decomposition - a été réalisée avec succès lorsque les deux contraintes cinématiques (inextensibilité et incompressibilité) ont été introduites en utilisant une formulation de pénalité pour contourner les contraintes LBB. Cependant, une telle formulation rend difficile le calcul des tractions des fibres et des forces de compression, une quantité importante d'intérêt nécessaire pour les caractérisations rhéologiques. Dans ce travail, l'ancienne formulation de pénalité est remplacée par une formulation mixte qui utilise deux multiplicateurs de Lagrange, tout en traitant les conditions de stabilité de la LBB dans un cadre séparé, questions jamais abordées jusqu'à présent.

Contents

CHAPTER 1

Introduction

Abstract   Simulation Based Engineering Science (SBES) has brought major improvements in optimization, control and inverse analysis, all leading to a deeper understanding in many processes occuring in the real world. These noticeable breakthroughts are present in a vast variety of sectors such as aeronautic or automotive industries, mobile telecommunications or healthcare among many other fields. Nevertheless, SBES is currently confronting several difficulties to provide accurate results in complex industrial problems. Apart from the high computational costs associated with industrial applications, the errors introduced by constitutive modeling become more and more important when dealing with new materials.

Concurrently, an unceasingly growing interest in concepts such as Big-Data, Machine Learning or Data-Analytics has been experienced. Indeed, this interest is intrinsically motivated by an exhaustive development in both data-acquisition and data-storaging systems. For instance, an aircraft may produce over 500 GB of data during a single flight. This panorama brings a perfect opportunity to the so called Dynamic Data Driven Application Systems (DDDAS), whose main objective is to merge classical simulation algorithms with data coming from experimental measures in a dynamic way. Within this scenario, data and simulations would no longer be uncoupled but rather a symbiosis that is to be exploited would achieve milestones which were inconceivable until these days. Indeed, data will no longer be understood as a static calibration of a given constitutive model but rather the model will be corrected dynamicly as soon as experimental data and simulations tend to diverge. Several numerical algorithms will be presented throughout the manuscript whose main objective is to strengthen the link between data and computational mechanics. The first part of the thesis is mainly focused on parameter identification, data-driven and data completion techniques. The second part is focused on Model Order Reduction (MOR) techniques, since they constitute a fundamental ally to achieve real time constraints arising from DDDAS framework.

Contents

1.1   Motivation

Mathematical modeling, computational algorithms, high performance computing constitute a discipline known as Simulation Based Engineering Science (SBES). It has brought the field of computer simulation to a new era where major improvements in optimization, multiscale modeling, control, inverse analysis have been achieved. Plenty of different numerical techniques have been developed throughout the history to improve the reliability of numerical simulations. It is worth to mention techniques as the finite element method, the finite difference or the finite volume method among many other approaches, constituting three of the main methodologies that are still widely employed nowadays. Even though they differ in many technical aspects, their common objective is to transform a given set of partial differential equations into a discrete system of algebraic equations which is perfectly understood by a computer.

Analyzing more is detail how a physical process is modeled, a classification between two different sets of equations can be done. The first set, coined balance or conservative laws, tries to preserve or equilibrate a given quantity throughout the domain as it is the case, for instance, of linear and angular momentum balance or conservation of energy. The second set, dubbed as constitutive equations, connects these balanced or preserved quantities with kinematic variables to render the solution unique. For instance, Hooke's law in solid mechanics or the plastic yield function in plasticity theory can be seen as examples of constitutive equations. The last set of equations have been used even before the invention of the differential calculus. Notably, Tycho Brahe's observations about planet's movements lead Johannes Kepler to formulate his three constitutive equations today known as the three Kepler's laws. Years after, Newton will generalize Kepler's laws with the formulation of the universal gravity law. Indeed, it is fair to reckon that constitutive equations have played a fundamental role in the modelization of most physical processes.

However, as the understanding of a given physical process goes deeper, its modelization from a mathematical point of view becomes more complex. This scenario can be appreciated in the solid mechanics framework where linear elasticity theory evolves to more complex situations such as non-linear solid mechanics or inelastic behaviours. Ilustrating this panorama, even though Hooke's law is widely employed in linear elasticity theory, the constitutive equation catalogue enlarges when considering hyper-elastic materials, i.e. Neo-Hookean, Saint Venant-Kirchhoff, Ogden, Mooney-Rivling, etc. The same effect can be observed when considering inelasticity, where plenty of different plasticity or damage models have been formulated to approximate the real behaviour of the material. Another scenario whose modelization becomes rather complicated is a complex fluid, as for example the case of foams flows. The numerical modelling of forming processes involving the flow of foams requires taking into account the different problem scales. Thus, in industrial applications a macroscopic approach is suitable, whereas the macroscopic flow parameters depend on the cellular structure: cell size, shape, orientation, etc. Moreover, the shape and orientation of the cells are induced by the flow. A fully microscopic description remains useful to understand the foam behaviour and the topological changes induced by the cell elongation or distortion, however, from an industrial point of view, microscopic simulations remain challenging to address practical applications involving flows in complex 3D geometries. In practice, viscoelastic macro constitutive models, where the foam microstructure is represented from suitable microstructure descriptors, the macroscopic flow kinematics govers the process. The derivation of such constitutive behaviours is detailed in Appendix A, certainly, it reinforced the suspicions about the complexity related to new material modeling.

Hence, it is reasonable to think that the phenomenological nature of constitutive equa-

tions may introduce some modeling error, leading to mismatches between simulations and experimental results. Furthermore, this modeling error tends to increase naturally as soon as the complexity of the system increases. As a consequence, plenty of effort has been dedicated to reduce every source of error in the numerical simulations, not only the modeling error but the discretization errors as well. Indeed, an ideal scenario where experiments and numerical simulations were in perfect agreement would allow to go beyond the current design limits. Not only because the exploration of new prototypes would be quantitatively cheaper but also because the knowledge inside the numerical simulation is by far richer than the data collected in experimental sensors. Unfortunately, this idyllic world does not exist these days, thus, numerical tools should be adapted accordingly to face this problematic. This transformation of standard numerical simulations is claimed in [Chinesta *et al.* 2018]. Traditionally, simulations are mainly static in the sense that constitutive equations are calibrated before doing the simulation without being able to modify them dynamically as soon as experimental data tends to differ from simulations. Nowadays, the situation is radically different, data is much more abundant (and accurate) than existing models, and a new paradigm is emerging in engineering sciences and technology, the so called Dynamic Data Driven Application Systems (DDDAS) [Darema 2005]. For instance, high-energy physics experiments produce around 1Pb of data per day, while in 2012, 162000 papers were published in materials science and engineering journals. These facts point clearly in the direction that data is becoming more and more abundant whereas the catalogue of possible materials is enormously enlarging. Thus, a coherent and efficient merging between data and standard simulation tools constitutes the current via of investigation for plenty of research centers.

This unprecedented possibility of directly determine or enrich knowledge from data or, in other words, to extract models from experiments in a automated way, is being followed with great interest in many fields of science and engineering. While almost classical in other domains of science like economics, sociology, etc., big data has arrived with important delay to the field of computational mechanics. It is worth noting that, in our field, the amount of data available is very often no so big, and therefore we speak of data-driven techniques instead of big-data techniques. For instance, the possibility of fitting the available data to a particular set of models has been recently explored in [Brunton *et al.* 2016, Kaiser *et al.* 2018]. Willcox and coworkers, on the contrary, have established a strategy that allows to construct reduced-order models from data, by inferring the full-order operators without the need to construct them explicitly, nor having a direct knowledge on the governing models [Peherstorfer & Willcox 2016, Peherstorfer & Willcox 2015]. Closely related, Ortiz has developed a method that works without constitutive models, by finding iteratively the experimental datum that best satisfies conservation laws [Kirchdoerfer & Ortiz 2016]. More recently, the issue of compliance with general laws like the ones of thermodynamics has been also achieved, which is a distinct feature of data-driven mechanics [González *et al.* 2018b]. Other applications include the identification of biological systems [Mangan *et al.* 2016] or financial trading [Mann & Kutz 2016], to name but a few.

On the other hand, tools related to advanced clustering or classification techniques become crucial in many areas where models, approximation bases, parameters, etc. are adjusted depending on the latent space, which is a reduced space representing the underlying physics of the system [Liu *et al.* 2016]. They make possible to define hierarchical, goal-oriented modeling and ultimately define a new set of reduced coordinates carrying most part of the information. Machine Learning techniques [Lee & Verleysen 2007] are able to extract the manifold structure in which the solution of complex and coupled engineering problems is

living. Thus, uncorrelated parameters can be efficiently extracted from the collected data, coming either from numerical simulations or experiments. As soon as uncorrelated parameters are identified (constituting the information level), the solution of the problem can be predicted at new locations of the parametric space, by employing adequate interpolation schemes [González *et al.* 2015, Lopez *et al.* 2016b]. On a different setting, parametric solutions can be obtained within an adequate framework able to circumvent the curse of dimensionality for any value of the uncorrelated model parameters [González *et al.* 2016]. It is at this point important to note that interpolation is a tricky issue when the solution defines a slow manifold embedded in the high dimensional space. In that case to define safe interpolation one must proceed by interpolating on the manifold. Farhat was one of the first to claim this necessity [Amsallem & Farhat 2008]. In[Millán & Arroyo 2013, González *et al.* 2016, Lopez *et al.* 2016a, Meng *et al.* 2018] manifold learning was used (using nonlinear dimensionality reduction strategies [Roweis & Saul 2000, Lee & Verleysen 2007, Maaten & Hinton 2008]) to extract latent parameters and the structure of the solution manifold in order to define accurate interpolations. It has also been employed for the construction of reduced models operating on the manifold, to define parametric solutions on it, or simply to define successfully data-driven computational mechanics applications. In these last cases, traditional constitutive equations were replaced by a manifold consisting of collected data.

Similarly, Model Order Reduction (MOR) is acquiring an utmost importance for simulation-based engineering. These techniques allow to efficiently solve complex mathematical models, thanks to the use of adapted approximation bases to describe their solutions. Among the numerous existing MOR techniques, Proper Orthogonal Decomposition (POD), Proper Generalized Decomposition (PGD) and Reduced Basis (RB) are largely considered in a variety of applications [Chinesta *et al.* 2017]. Indeed, MOR techniques could be seen as a natural ally to explore new routes where data, models and simulations constitute a solid framework to drive technological breakthroughts.

Proper Orthogonal Decomposition is a general technique to extract the most significant characteristics of a system's behavior and to represent them in a set of optimal "POD basis vectors". These basis vectors provide an efficient (typically, low-dimensional) representation of the essential features of the system behavior, which has proven useful in a variety of ways. The most common use is to project the solution of the governing equations onto the reduced-order subspace defined by these POD basis vectors. This yields an explicit POD reduced model that can be solved instead of the original system. The POD basis can also provide a low-dimensional description on which to perform parametric interpolation, infill missing or "gappy" data, and perform model adaptation. There is an extensive literature and POD has seen broad application across fields. Some review of POD and its applications can be found in [Volkwein 2001, P. Benner 2016].

Reduced Basis techniques employ an approximation basis constructed by combining a greedy algorithm and *a posteriori* error indicators. As for the POD, the Reduced Basis method requires some amount offline work, but then the reduced basis model can be used online for solving different models with control of the solution accuracy, because the availability of error bounds. When the error is unacceptably high, the reduced basis can be enriched by invoking a greedy adaption strategy. Useful review works on the subject are [Patera & Rozza 2007, Rozza *et al.* 2008, Quarteroni *et al.* 2011].

Finally, there exist techniques based on the use of separated representations, at the heart of the so-called Proper Generalized Decomposition (PGD) methods [Chinesta & Cueto 2014]. Such separated representations are considered when solving at-hand partial differential equations by employing procedures based on the separation of

variables. They were already considered in quantum chemistry to approximate multidimensional quantum wave-functions, e.g., Hartree-Fock and post-Hartree-Fock methods. In the 80s, Pierre Ladeveze proposed the use of space-time separated representations of transient solutions involved in strongly nonlinear models, defining a non-incremental integration procedure [Ladeveze 1989, Ladeveze 1999]. Later, separated representations were employed for solving multidimensional models suffering the so-called curse of dimensionality [Ammar *et al.* 2006, Ammar *et al.* 2007] as well as in the context of stochastic modeling [Nouy 010]. Then, they were extended for separating space coordinates making possible the solution of models defined in degenerated domains, e.g. plate and shells [Bognet *et al.* 2012] as well as for addressing parametric models where model parameters were considered as model extra-coordinates, enabling the offline calculation of the parametric solution. This precomputed solution can be viewed as a metamodel or a computational vademecum, to be used online for real time simulation, optimization, inverse analysis and simulation-based control (see [Chinesta *et al.* 2013b] for a recent review). Some recent reviews concerning the PGD can be found in [Chinesta *et al.* 2010, Chinesta *et al.* 2011b].

In the light of this data revolution, it is easy to understand the reasons why the concept of Industry 4.0 is gathering a lot of interest recently. The possibility of creating smart processes where data plays a major role in order to enhance efficiency is a rather appealing option. This improvement would be achieved thanks to the correct merging of the collected data and the process to be optimized. This collected data may be, for instance, quantitative variables such as the historical evolution of temperatures, pressure or displacements or qualitative ones like manufacturers, materials, name of the operator, etc. This dataset will feed the process itself determining the actions to be taken in real time. However, as this Industry 4.0 is still an emerging idea, several aspects must be treated and improved before transforming it into a reality. At first glance, this concept of smart factory may seem very problem dependent, or in other words, each problem may depend very much on the process to be optimized. However, the main drawbacks associated to work with data could be classified as:

- *Data Rationalization*. Big-Data rationale promotes to collect a data set as big as possible. However, this data excess may be counter-productive when trying to unveil hidden relationships as the extraction of those underlying relationships will be much more expensive. Therefore, it is important to keep always in mind that Big-Data should evolve to Smart-Data, where questions such as why-when-where is data going to be collected are treated before actually collecting it. The formulation of this kind of questions may derive into a considerable problem simplification by selecting the most important input and output variables which are controlling the process.

- *Data Completion*. Despite of the fact that Big-Data inherently involves working with a huge amount of data. The truth is that sometimes the amount of accesible data is not as rich as required. This degeneration into the low-data limit may be caused by several reasons. For instance, a given data set may become low or sparse data when considering high dimensional spaces, arising when considering all variables affecting the process. Another low-data case may appear when considering data provided by sensors since they are placed at specific locations of the process. As a result, the state variables will not be defined everywhere in the domain but only at certain sensors locations.

- *Fast Numerical Algorithms*. New algorithms intending to combine data and standard simulations processes are embedded in the so-called Dynamic Data Driven Application Systems (DDDAS). Indeed, the word Dynamic incites such data corrections to

be done in real time, or at least, as dynamic as possible. It seems natural to make use of MOR methodologies to speed up the algorithms as much as possible. They can be used as an efficient tool to explore high-dimensional spaces, arising, for instance, when dealing with parameter identification algorithms.

- *Consistent Data Merging.* Using data is convenient, unfortunately, there are situations in which the available data does not satisfy certain conditions *per se*. For instance, we should be reticent or at least aware of situations such as a non-positive conductivity identification from a temperature-heat flux profile just like a non-positive definite elasticity tensor constructed from a displacement-force data set. These degenerate situations may lead to perspectives where data is no longer a natural substitute of existing models, but rather an ally to enrich existing models preserving the thermodynamic consistency [Gonzalez *et al.* 2019].

The contributions presented in this thesis address certain difficulties presented in the above list, developing numerical algorithms which strengthen the link between data and simulation based engineering science. More details about each contribution are specified in section 1.2.

1.2 Structure of the thesis

As mentioned in section 1.1, there is an unceasingly growing interest in merging fields like Data-Science and Science Based Simulation Engineering to derive knowledge from information and ultimately transform this knowledge into decision making. Most decisions to be taken into a industrial environment can be made from a new kind of (artificial) intelligence that, more than based on mathematical expressions, are based on data, via data mining and data analytics. This thesis is organized in two differentiated parts, addressing the problems presented in section 1.1. Briefly, part I constitutes a first step towards an efficient and coherent merging of data science and computational mechanics fields. Part II is focused on MOR techniques, since they constitute a fundamental ally to achieve real time constraints arising from DDDAS framework. The overall structure of the thesis is given by:

- ***Part I:*** *First steps towards an efficient merging of data sience and computational mechanics*

  - ***Chapter 2:*** *Data-Driven Techniques*
  - ***Chapter 3:*** *Data Completion Techniques*

- ***Part II:*** *Advances on Model Order Reduction Techniques*

  - ***Chapter 4:*** *Multi PGD based on Partition of Unity Method*
  - ***Chapter 5:*** *Applications of PGD In-Plane-Out-Of-Plane Separations*

1.2.1 *Chapter 2: Data-Driven Techniques*

In this chapter, the usage and identification of constitutive relationships is treated. The constitutive relationship is no longer understood as an equation in its closed format, but as a cloud of points or data set. A first attempt to merge this constitutive data base with a numerical scheme is done in [Kirchdoerfer & Ortiz 2016, Kirchdoerfer & Ortiz 2017] for elasticity framework. Therein, the solid is treated as a set of 1D bars which are calibrated with standard stress/strain curves. These stress/strain curves carry all constitutive material

information. Afterwards, an iterative scheme between these experimental curves and the linear momentum balance is solved. Certainly, Data-driven simulation consists of directly employ data in order to perform numerical simulations. These simulations will employ universal laws while minimizing the need of explicit, often phenomenological, models. This approach is especially interesting when considering complex engineered materials (meta-materials), for which constitutive relations become hard to write, because there are (too) many possible designs, and the intimate nature of most of them remains inaccessible and/or confidential.

Even though conceptually speaking either a set of points representing the constitutive behaviour or the actual equation in its closed form carry the same information, transforming it into a data base allows to extract the most relevant features inside the data collection. Machine and manifold learning techniques, and more specifically non-linear dimensionality reduction, as for example locally linear embedding (LLE), kernel-PCA (the nonlinear counterpart of principal component analysis —PCA), referred as k-PCA, local-PCA, among many other choices, allows us to remove correlations in data [Lee & Verleysen 2007, Polito & Perona 2001, Tenenbaum *et al.* 2000, Wang 2012]. Such data, free of correlation, constitutes the real information, often very limited when compared with the big data from which it was extracted. In many models, the extraction of uncorrelated parameters remains a tricky issue. It is the case of parameters describing microstructures or shapes for example, often referred to as latent parameters. As soon as the uncorrelated parameters are extracted, two main options have been considered to the date: (i) when a new case, not included in the data, must be analyzed, its solution is simply interpolated on the manifold (constructed from the training data) from its closest neighbors [Lopez *et al.* 2016a] so that decisions can be taken in real time; and (ii) an explicit parametric solution could be constructed by using the just extracted uncorrelated parameters so that it could be particularized in real-time [González *et al.* 2016].

Section 2.1 treats the usage of the constitutive data set in the elasticity framework. The solid is no longer understood as a set of 1D bars but rather as a continuum solid where stress and strains, both second order tensors, are connected by a fourth order tensor called the elastic modulus. As a consequence the constitutive data set has twelve coordinates in 3D, six to define the stress and other six to define the strain states assuming it's symmetry, where manifold learning techniques can be applied. Afterwards, several techniques are presented to merge the constitutive data set together with the linear momentum balance to be consistent with standard finite element algorithms. This section will not address all the technical difficulties related to data generation or obtention from adequate experiments. On the contrary, we develop a method in which this stream of data plays the role of a constitutive equation, without the need of a phenomenological fitting to a prescribed closed model.

Section 2.2 addresses one of the current bottlenecks of Data-Driven simulations, the huge amount of required data, some of them inaccessible from the nowadays testing facilities. Indeed, it is hard to explore the entire stress/strain space by means of homogeneous experiments where only limited stress states are reachable, i.e. uniaxial or biaxial loadings. Such difficulty can be circumvented in many cases, and in all cases alleviated, by considering complex (or heterogeneous) tests, where plenty of different stress/strain states coexist inside the material. Then, a data-driven inverse approach allows to extract a big amount of data from the complex experiments. A subtle alliance of testing machines, devices for collecting data and powerful computers for treating these huge amount of data in a variety of ways (machine and deep learning) allows to unveil the constitutive manifold behind the experiment.

Section 2.3 abords another application within the data-driven rationale. Certainly, upscaling behaviors of heterogeneous microstructures to define macroscopic effective media is of major interest in many areas of computational mechanics, in particular those related to materials and processes engineering. In this work, we explore the possibility of defining an effective macroscopic behavior from a data set based on microscopic calculations, and then use it directly for efficiently performing manifold-based simulations at the macroscopic scale. We consider in this work upscaling of non-Newtonian flows in porous media, and more particularly the ones involving short-fibre suspensions. On the other hand, when considering flows of simple or complex fluids in porous media, the situation becomes difficult because the physics encountered at the micro-scale differ from those postulated at the macro-scale. Thus, the micro-scale flow is governed by a Stokes flow problem, that becomes non-linear as soon as the fluid viscosity depends on the rate of strain. In the case of suspensions of rods (fibres, micro-fibres, nano-fibres or nanotubes), the micro-scale flow is governed by an anisotropic Stokes problem wherein viscosity is locally highly anisotropic. It is meaningless, however, to homogenize the microscopic flow parameters in order to define an effective viscosity at the macroscopic scale. Thus, the most valuable route consists in considering as much as possible viscous models that are more or less complex to capture the main fluid and flow features at the microscopic scale, and from them obtaining (by upscaling [Bohm 2009, T. Zohdi 2005]) an effective Darcy permeability at the macroscopic scale to be given as input to conventional simulation software.

1.2.2 Chapter 3: Data Completion Techniques

As it was detailed in section 2.2, a latent problem involving data driven applications is the data generation, as sometimes it is inaccessible or expensive to obtain it. Therefore, developping data completion algorithms where response surfaces are created based on an existing data set is of vital importance. The problem with high dimensional systems is that data in these systems is often sparse (due precisely to the high dimensional nature of the phase space) while the system very frequently has, on the contrary, low dimensional features.

MOR techniques improved traditional strategies based on DoE (Design of Experiments), that allowed defining metamodels, surrogate models or response surfaces. In these cases, experiments or expensive computational solutions are performed for a sampling of possible states of the system, from which a simplified model linking inputs to outputs is elaborated. The main difficulties associated to this procedure concern the best sampling strategy and the most adequate interpolation scheme for making prediction everywhere in the design space from the only knowledge of the few analyzed scenarios. Latin hypercube and Kriging are two usual responses to these questions. However, other questions remain open, such as model verification (error estimation and bounds) as well as the definition of adaptive strategies able to reduce such error locally or globally. Even if there is a panoply of proposals and applied strategies, most of them are problem-dependent and fail to be robust and reliable. As just imentioned, model order reduction established routes to achieve similar goals while circumventing the main issues just indicated, to finally define a "numerical or graphical handbook", constructed offline and efficiently used online for robust design purposes.

Even if, as just indicated, MOR technologies facilitate better approaches, their main drawback is that they often remain too intrusive. Nowadays, the most recent works concerning MOR techniques focus on non-intrusive algorithms. However, from a pragmatic point of view, all these proposals remain less direct than usual DoE methodologies, the last simply consisting of evaluating the model at different points in the design space by using standard commercial solvers (adapted to the problem at hand) and then simply interpolat-

ing these solution to any other point. Despite the conceptual difficulties just referred, the procedure is very simple and attracted the favor of engineers, designers and practicians.

Thus, the big picture could be formulated as follows: could direct sampling lead to a robust and reliable parametric solution?

The answer to this question has been traditionally addressed in a variety of ways. First, response surface based methodologies (e.g. [Breitkopf *et al.* 2005] and references therein) proposes a sort of adaptive procedure when the model is refined by zooming-in when solution approaches to the optimal solution with respect to a given couple model / optimization criterion. However, such a procure requires an amount of online computation because the fine representation is not *a priori* available and it must be constructed online during the optimization process.

Other possibility consists in reconstructing the unknown solution everywhere from the only knowledge of the calculated scenarios by making use of adequate interpolations: polynomials or POD-based modes (inspired from the gappy-POD formulations [Everson & Sirovich 1995]). Reduced Basis performs this job offline: it extracts a basis in which the model solution is projected. This projection is then solved online. However, one could imagine using directly the offline computed solution for interpolating it everywhere. The main advantage of this procedure lies in the fact that in the RB framework the sampling points (as indicated above) are determined from an adequate "a priori"—or more generally "a posteriori"—error indicator, defining a sort of greedy strategy that samples the space at (almost) optimal points. The main drawback is that, very often, the definition of those error indicators requires some deep knowledge of the considered model and it is not evident for many complex engineering problems. It is important to note that the fact of extracting the basis for projecting the problem solution and solving online the reduced problem offers higher precision that the option of directly interpolating the sampled solutions.

Close to the methodology just described, Borzacchiello et al. [Borzacchiello *et al.* 2017a, Borzacchiello *et al.* 2017b] proposed the use of hierarchical approximation bases, enabling that at each level of representation only the contribution from the previous level to the present one must be calculated. This strategy allows defining simple error indicators and adaptively refine the parametric domain. Similar strategies, all them inspired of sparse-grids methodologies [Bungartz & Griebel 2004] can be combined with the use of wavelet representations in order to profit their inherent multi-resolution properties, that provides natural error indicators associated with the weights of the wavelet coefficients at each representation level [Leon *et al.* 2018]. This techniques allows even addressing multi-parametric models in a moderate number of dimensions. However, in high dimensional settings, all of the just mentioned techniques fail to identify the nature of the system due precisely to the curse of dimensionality. A recent alternative for such a system could be Topological Data Analysis (TDA), which is based on the employ of algebraic topology and the concept of *persistent homology* [Epstein *et al.* 2011]. A sparse version of this technique also exists [Brunton *et al.* 2016].

That being said, an efficient data completion technique within data-driven framework should meet the following requirements:

- *Non-structured data set*: this characteristic provides versatility to the method. Indeed, when evaluating the response surface requires a lot of computational effort, recycling previous evaluations of the response surface, which do not coincide with a given structure of the data, may be very useful. Moreover, having algorithms which are capable of working with non-structured data-sets is of vital importance for certain applications where parametric space locations are hard to explore. For instance,

exploring uniaxial loading states in the constitutive manifold is easier than biaxial states, and even easier than triaxial states.

- *High-dimensionality robustness*: most data completion techniques, as the ones based on triangularization, suffer when dealing with multidimensional data just because of the fact that a high dimensional mesh has to be generated. Nevertheless, separation of variables could be an appealing technique to circumvent, or at least alleviate, the problem of addressing high-dimensional spaces.

- *Curse of dimensionality*: all previous techniques suffer when dealing with high dimensional data. For instance, the SSL needs 2^D sampling points just to reach the first level of approximation. Thus, when dealing with high dimensional data ($D > 10$ uncorrelated dimensions) plenty of sampling points would be required to construct a first approximation of a given response surface.

- *Low data limit*: the robustness of the data completion techniques where the amount of data is limited is crucial. Indeed, situations in which the cost of enlarging the data set is unaffordable will require a numerical technique which is able to provide a reasonable response surface based on the given sparse data set.

Section 3.1 explains a novel data completion technique called sparse Proper Generalized Decomposition (sPGD). It combines the Proper Generalized Decomposition (PGD) rationale, where separation of variables is used to circumvent, or alleviate, the curse of dimensionality together with a collocation procedure coincident with the unstructured data set. By doing that, no information out of the data set is required, thus, constructing an efficient response surface based on the available data.

Section 3.2 portraits three different applications of industrial interest in which the sPGD technique explained in section 3.1 has been applied. The first application in collaboration with *Gestamp* concerns the automotive car industry, where the geometrical shape of a component inside the car is optimized to guarantee the safety of the passengers. The second application in collaboration with *ESI Group* aims to correct existing plasticity models by means of observable quantities i.e. strain, displacement synthetic measures. The third application links the geometrical parameters defining a composite material surface with its performance in the Automated Place Tapement (ATP) process. Indeed, geometrical parameters such as curvature or roughness are extracted from real composite materials usign a profilometer. Several clustering and data completion techniques, the sPGD amongst them, are use to link the former parameters with the performance of the composite material.

Section 3.3 extends the concept of sparse identification techniques, already introduced by [Brunton *et al.* 2016] to identify the model hidden in a data set, to other applications within the computational mechanics framework. This concept has been widely applied in other fields like image compression, by evoking the concept of single pixel camera. This single pixel camera or compressed sensing claims to be able to reconstruct a given image just by knowing the information at few pixels. The same rationale is applied to reconstruct a parametric response surfaces just like the possibility of applying the methodology to time evolving problems.

1.2.3 *Chapter 4: Multi PGD based on Partition of Unity Method*

Model order reduction techniques (MOR) become very useful within the Dynamic Data Driven Application System framework where decision making has to be done in a dynamic way. In recent years, MOR techniques have shown that a minimum number of

carefully-chosen degrees of freedom are enough for an accurate solution of PDEs, alleviating the cost of standard FEM discretizations. Instead of choosing general-purpose piecewise polynomials as basis functions, MOR techniques construct *ad hoc* basis functions following different techniques. For instance, Proper Orthogonal Decomposition (POD) [Karhunen 1946, Loève 1963, Park & Cho 1996, Meyer & Matthies 2003] constructs an efficient basis from a set of precomputed snapshots of the full-order PDE solution,

$$u(x,t) = \sum_{i=1}^{N} \alpha_i(t)\phi_i(x), \qquad (1.1)$$

where, very much like in the finite element context, α_i are a set of time-dependent coefficients that evolve in time, and $\phi_i(x)$ are time-independent basis functions obtained by some statistical treatment of the system snapshots. For instance, many MOR techniques employ the most energetic eigenfunctions of the snapshot autocorrelation matrix to construct these $\phi_i(x)$. These play a similar role to the finite element shape functions, albeit they are global instead of local. Other techniques, such as Reduced Basis methods, for instance [Patera & Rozza 2007] [Quarteroni *et al.* 2011] [Rozza 2014], employ some snapshots of the full-order solution as basis for the approximate solution of the system. These snapshots are calculated in a greedy fashion, at time (or parameter) instants at which the error in the approximation is maximal.

It is important to note that Eq. 1.1 constitutes in fact a separated or affine expression of the solution—note the space-time separation, which also holds in finite element approximations—. This methodology has proven to be very effective from a wide variety of high dimensional problems ranging from the resolution of Fokker-Planck equation, to patient-specific liver responses [Mena *et al.* 2015] [González *et al.* 2015], structural dynamics [Gonzalez *et al.* 2014], computational rheology [Chinesta *et al.* 2011a] or, more generally, to any parametric problem that could be written in separate form [Pruliere *et al.* 010]. There are situations, however, when the solution is highly non-separable. In other words, the solution of the problem lives on a non-linear manifold. In this situation, what MOR techniques do is to project the solution on the tangent space to the manifold at a given (time or parameter) point [Amsallem & Farhat 2008]. This leads to poor approximation properties far from the tangency point, unless special techniques are chosen, as in [Niroomandi *et al.* 2010], for instance, where asymptotic expansions were used.

A way to alleviate this problem is to follow the same philosophy than non-linear-dimensionality reduction methods. For instance, Locally-Linear-Embedding (LLE) [Roweis & Saul 2000] tries to unveil the latent variables by means of imposing a local linear variation on the function, which will change from neighborhood to neighborhood. Another technique which deals with non-linear problems is the so-called kernel Principal Component Analysis (k-PCA) [Scholkopf *et al.* 1999]. In this particular case, the definition of an efficient kernel function allows to project the snapshots to high dimensional spaces (potentially, infinite dimensional) in which the solution manifold is flat. In this particular situation, standard interpolation techniques work well. A. Badías et al. [Badías *et al.* 2017] studied the case of a moving source in a transient heat transfer problem. The problem of the non-separability of the solution was circumvented by means of making a partition of the time domain, dedicating a different PGD for each partition. However, the imposition of the interface conditions will become a cumbersome task when dealing with partitions involving variables other than time.

Section 4.1 proposes a methodology where different PGDs are combined thanks to the introduction of FEM macro shape functions satisfying the Partition of Unity property [Babuska & Melenk 1996, Babuska & Melenk 1997]. These macro shape functions, namely

FEM piecewise polynomial functions, are responsible of controlling the overlap between the different PGDs, whitout needing to specify interface conditions between them. Moreover, the fact that the solution becomes locally more linear (also more separable) facilitates the task to each one of the PGDs, reducing drasticly the number of modes to achieve an accurate solution.

Section 4.2 extends the methodology proposed in section 4.1 to treat multi-scales problems from an efficient point of view. Many problems in computational mechanics present a multi-scale behaviour where the micro-scale effects influence the macro-scale ones and viceversa. Hence, the treatment of the different scales of the problem becomes very important to reach an accurate solution. Such a multi-scale behaviour appears, for instance, in the spatial domain when dealing with metamaterials whose structure is defined at the micro level. Several techniques have been applied throughout the history like the so-called homogeneization [Michel *et al.* 1999], defining the micro-scale constitutive model in terms of a representative volume that satisfies the Hill-Mandel principle [Feyel 1999]. This methology has been proven to be very effective in order to circumvent the prohibitive computational cost of methods like FEM^2. However, there is a strong hypothesis behind such homogeneization technique, requiring a clear separation of scales between macro and micro effects. Another way to handle these multi-scale effects is by using the variational multi-scale framework introduced by [Hughes *et al.* 1998], where the effect of the micro-scale into the macro-scale variables is introduced in a consistent way developping different weak forms associated to the macro and micro scales.

Another appealing multi-scale application is the transient problems. Standard time marching approaches require a suitable time step that captures the evolution up to the finest time scale. A vast variety of solutions have been proposed to circumvent this prohibitive time step.

D. Neron et al. [David & Ladeveze 2010] proposed a numerical algorithm based on the LATIN-PGD able to handle temporal and spatial multi-scale behaviours appearing in solid mechanic problems. However, selecting the temporal macro basis just like the interfacial degrees of freedom coupling different macro domains is very problem dependent. The scalability of the methodology is restored via an appropiate correction of the temporal basis based on the residual. F. Fritzen et al. [Fritzen & Leuschner 2013] also partitioned the time domain into subintervals, where a common reduced basis is applied for all subintervals. Afterwards, extra interface restrictions have to be imposed to ensure continuity of the primal variable and her time derivative. Hence, variables at both extremes of each macro interval are set to zero, plus an offset calculated from the previous macro interval. Several viscoelastic problems are solved for various cyclic loading with varying frequency. Y. Maday et al. [Maday & Turinici 2002] developped Pararreal algorithm which also attempts to solve a temporal evolution iterating between macro and micro domain partitions so that the initial conditions for each micro interval are given by the macro resolution of the problem. The algorithm is also highly parallelizable, making it very efficient for either long time simulations or temporal problems involving many spatial degrees of freedom. A. Ammar et al. in [Ammar *et al.* 2011] proposed an efficient technique based on the separation of variables, where the time domain was partitioned into discretes subdomains following the standard PGD rationale [Chinesta *et al.* 2013a, Niroomandi *et al.* 2010], [Mena *et al.* 2015], [Pruliere *et al.* 010]. However, special attention had to be paid at the interfaces between discrete subdomains, where continuity was imposed by means of Lagrange multipliers.

Indeed, section 4.2 proposes a methodology in which macro shape functions that respect the partition of unity allowed the method to combine different PGDs in a consistent manner. Moreover, the overlap between PGDs automaticly solves transition between different scales.

1.2.4 *Chapter 5:Applications of PGD In-Plane-Out-Of-Plane Separations*

Many mechanical systems and complex structures involve plate and shell parts whose main particularity is having a characteristic dimension (the one related to the thickness) much smaller that the other ones (in-plane dimensions). Introducing appropriate kinematic and mechanic hypotheses allows the reduction of the general 3D mechanical problem to a 2D involving the in-plane coordinates. This was the route employed for deriving beam, plate and shell theories in solid mechanics, that were later extended to many other physics, like flows in narrow gaps, thermal or electromagnetic problems in laminates, among many others. Yet when physics is applied on degenerated domains, like plates or shells, and no acceptable simplifying hypotheses are available for reducing their complexity to 2D, fully 3D solutions seem compulsory. This is, for instance, the case when considering the progressive dynamic damage of composite laminates where a rich through-the-thickness description could be extremely valuable, among many other scenarios in which a fully 3D formulation is retained.

When such 3D models become compulsory the approximation of the different fields could imply thousands of nodes distributed along the thickness direction, and consequently millions of nodes to represent the fields in the part. Today, the solution of such rich 3D models remains intractable despite the impressive progresses reached in modeling, numerical analysis, discretization techniques and computer science during the last decades. Standard mesh-based discretization techniques fail because the excessive number of degrees of freedom involved in the fully 3D discretization. Indeed, very fine meshes are required in the thickness direction (despite its reduced dimension) and many times also in the in-plane directions to either avoid too distorted meshes or also because some processes (e.g. microwaves) require fine in-plane representations.

In order to alleviate the associate computational complexity, authors proposed few years ago to separate the physical space. Thus, an in-plane-out-of-plane decomposition was proposed for solving in plate 3D flows occurring in RTM – Resin Transfer Moulding – processes [Chinesta *et al.* 2011a], then for solving elasticity problems in plates [Bognet *et al.* 2012] and shells [Bognet *et al.* 2014]. In those cases the full 3D solution was obtained from the solution of a sequence of 2D problems (the ones involving the in-plane coordinates) and 1D problems (the ones involving the coordinate related to the plate thickness). It is important to emphasize the fact that these approaches are radically different from standard ones. We propose a 3D solver able to compute the different unknown 3D fields without the necessity of introducing any hypothesis. The most outstanding advantage is that 3D solutions can be obtained with a computational cost characteristic of standard 2D solutions.

This chapter is meant to present some novel applications of the In-Plane-Out-Of-Plane separated representation.

Section 5.1 aims to separate the spatial dimentions within the electromagnetic framework. The main motivation behind this numerical study resides on the highly demanding energy and time required by an autoclave. Many composite manufacturing processes imply the use of autocalve to ensure after forming the simultaneous application of temperature and pressure to achieve both consolidation and low porosity. The high cost of this processes in terms of energy and time has recently lead the research of other technologies to subsitute autoclaves. Among them, microwave heating is being considered as a nice candidate for speeding-up manufacturing processes. Here, depending on the electromagnetic properties of the reinforcement two thermal sources can coexist: dielectric losses and induction, the last related to electrically conductive reinforcements (e.g. carbon fibres). In the context of electromagnetism the situation was similar to the one encountered in structural mechanics

as discussed in [Swanson & Hoefer 2003], where 2D cross-section, 2.5D planar solvers and 3D arbitrary-solvers were discussed. Waveguides simulators widely employed 2.5D formulations where one of the dimensions was eliminated by assuming a particular evolution of the electromagnetic field in that direction [Chew & Nasir 1989, Pan & Tan 1997]. 2.5D formulations have been also extensively considered for analyzing printed circuits [E.X. Liu 2016]. Microscopic, mesoscopic and macroscopic models have been proposed and widely considered in engineering applications, in particular in those involving laminates. Many works concerned the microscopic homogenization of electromagnetic systems, in particular composite materials [Wasselynck *et al.* 2010]. At the mesoscopic scale laminates were addressed in view of defining macroscopic approches. Three main routes were considered: (i) laminate homogenization [Niyonzima *et al.* 2013] [Gyselinck *et al.* 2016]; (ii) monolayer and multi-layer shell elements were proposed in [Bensaid *et al.* 2005] [Bensaid *et al.* 2006], where the shell impedance was analytically derived by assuming a number of simplifying hypotheses and then coupled with the electromagnetic problem solution in the external domain in order to obtain the electromagnetic fields on the shell surfaces, and from them calculating all the physics inside (it is important to note that the assumed hypotheses could, in some complex circumstances, prove defective); (iii) special elements were proposed in [Bui *et al.* 2016] for meshing laminates when addressing explicitly the solution inside, however, for finely representing boundary layers inside the plies the computing complexity due the required mesh resolution could become excessive. That is the reason why the separation of spatial dimensions seems suitable for this particular case.

Section 5.2 introduces a new hybrid explicit/implicit in-plane-out-of-plane separated representation for dynamic problems defined in plate-like domains that computes efficiently 3D solution and where the stability constraints are exclusively determined by the coarser in-plane discretizations. It exists a vast literature on structural dynamics, covering different discretization techniques and time integration procedures [Sun *et al.* 2000]. When considering an implicit analysis, solution at each time step needs some iterations to enforce equilibrium. Contrary to implicit schemes, explicit ones do not require iteration as the nodal accelerations are solved directly, and from which velocities and displacements are calculated by simple integration. The main handicap of explicit simulations is that the time step must verify the stability condition, decreasing with the element size. On the contrary implicit elastodynamic integrations become unconditionally stable, that is, there is not a limit in the time step to be considered in what concerns stability. Thus, implicit time steps are generally several orders of magnitude larger than the ones considered in explicit time integrations. However, implicit integration requires the solution of linear systems several times at each loading step when addressing nonlinear models. On the other hand, since explicit techniques do not require the inversion of a matrix, they easily address non-linearities (contact or material non-linearities). In [Hughes & Liu 1978] a hybrid schema was proposed that considers the domain composed of two parts in which explicit and implicit time integrations apply. In this work we introduce a new hybrid explicit/implicit in-plane-out-of-plane separated representation for dynamic problems defined in plate-like domains that computes efficiently 3D solution and where the stability constraints are exclusively determined by the coarser in-plane discretizations.

Section 5.3 focuses on thermoplastic composites since they are preferred structural materials due to their excellent damage tolerance properties, shorter manufacturing cycles and ease of weldability. One of the precursor material to fabricate thermoplastic composite parts is an unidirectional (UD) prepreg which consists of aligned continuous fibers pre-impregnated with thermoplastic resin. In their melt state, UD prepreg can be viewed as inextensible fibers surrounded by an incompressible viscous matrix, and hence can be

modeled as a transversally isotropic fluid [Spencer 2000]. These UD laminates are usually stacked in desired orientations to create a composite laminate. Indeed, an application of the in-plane-out-of-plane separated representation to squeeze flows of continuous fiber laminates is presented. More in detail, the ply constitutive equation is modeled as a transversally isotropic fluid, that must satisfy both the fiber inextensibility as well as the fluid incompressibility. When the laminate is squeezed, the flow kinematics exhibits a complex dependency along the laminate thickness requiring a detailed velocity description through the thickness. In [Ghnatios *et al.* 2016] the solution making use of an in-plane-out-of-plane separated representation within the PGD – Poper Generalized Decomposition – framework was successfully accomplished when both kinematic constraints (inextensibility and incompressibility) were introduced using a penalty formulation for circumventing the LBB constraints. However, such a formulation difficults the calculation on fiber tractions and compression forces, an important quantity of interest required in rheological characterizations. In this work the former penalty formulation is substituted by a mixed formulation that makes use of two Lagrange multipliers, while addressing the LBB stability conditions in a separated framework, questions never addressed until now.

1.3   Articles in Journals

These thesis works have generated the following publications:

1. R. Ibáñez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta, *A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity.*Archives of Computational Methods in Engineering, **25**, 47-57, 2018.

2. R. Ibáñez, D. Borzacchiello, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladeveze, F. Chinesta, *Data-Driven non-linear elasticity: constitutive manifold construction and problem discretization.* Computational Mechanics, **60**, 813-826, 2017.

3. R. Ibáñez, A. Scheuer, E. Lopez, E. Abisset-Chavanne, F. Chinesta, R. Keunings, *From elastic homogeneization to upscaling of non-Newtonian fluid flows in porous media.* International Journal of Material Forming, **1**, 1-11, 2017.

4. R. Ibáñez, E. Abisset-Chavanne, A. Ammar, D. Gonzalez, E. Cueto, A. Huerta, J. L. Duval, F. Chinesta, *A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition.*Complexity, **12**, 1-11, 2018.

5. V. Limousin, X. Delgerie, E. Leroy, R. Ibáñez, C. Argerich, F. Daim, J. L. Duval, F. Chinesta, *Model Order Reduction and Machine Learning Techniques in Advanced Structural Mechanics. Application to Crash Simulation.* To be submitted into Mechanics & Industry.

6. R. Ibáñez, E. Abisset-Chavanne, D. Gonzalez, J. L. Duval, E. Cueto, F. Chinesta, *Hybrid constitutive modeling: data-driven learning of corrections to plasticity models.* International Journal of Material Forming, DOI:10.1007/s12289-018-1448-x, 1-9, 2018.

7. C. Argerich, A. Leon, R. Ibáñez, A. Barasinski, E. Abisset-Chavanne, F. Chinesta, *Prediction of composites behavior undergoing an ATP process through data-mining.* AIP Conference Proceedings, DOI: 10.1063/1.5034817, 2018.

8. R. Ibáñez, E. Abisset-Chavanne, A. Ammar, E. Cueto, J. L. Duval, F. Chinesta, *Some applications of compressed sensing in computational mechanics.* Submitted to Computational Mechanics.

9. R. Ibáñez, E. Abisset-Chavanne, F. Chinesta, A. Huerta, E. Cueto, *A local, multiple Proper Generalized Decomposition based on the Partition of Unity*. Submitted to International Journal Numerical Methods for engineering, 2018.

10. R. Ibáñez, A. Ammar, E. Cueto, A. Huerta, J. L. Duval, F. Chinesta, *Multi Scale Proper Generalized Decomposition based on the Partition of Unity*. Submitted to International Journal Numerical Methods for engineering, 2018.

11. H. Tertrais, R. Ibáñez, A. Barasinski, C. Ghnatios, F. Chinesta, *On the Proper Generalized Decomposition applied to microwave processes involving multilayered components*. Mathematics and Computers in Simulation, **156**, 347-363, 2019.

12. G. Quaranta, B. Bognet, R. Ibáñez, A. Tramecon, E. Haug, 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 and Structures, **210**, 135-144, 2018.

13. R. Ibáñez, E. Abisset-Chavanne, F. Chinesta, A. Huerta *Simulating squeeze flows in multiaxial laminates: towards fully 3D mixed formulations*. International Journal of Material Forming, **210**, 653-669, 2017.

14. R. Ibáñez, A. Scheuer, E. Abisset-Chavanne, F. Chinesta, A. Huerta, *A simple microstructural viscoelastic model for flowing foams*. International Journal of Material Forming, 1-12, 2018.

15. G. Quaranta, C. Argerich, R. Ibáñez, J. L. Duval, E. Cueto, F. Chinesta, *From linear to nonlinear PGD-based parametric structural dynamics*. Comptes Rendus - Mecanique, 2019.

Part I

First steps towards an efficient merging of data science and computational mechanics.

Even though both data science and computational mechanics fields have gathered a lot of interest in the recent years, very little has been done to merge both fields. Plenty of benefits could be extracted when a perfect symbiosis would be fulfilled. Important steps in many scientific areas could be achieved, for instance, modeling of new materials would become easier as soon as data alleviates the weight of constitutive modeling. Moreover, the decision making characteristic of most industrial processes could be based on a new kind of artificial intelligence generated from data.

Therefore, the purpose of this part is to strengthen the link between data and standard simulation based engineering. This part is structured in two diffenent chapters. Chapter 2 is mainly devoted to novel data driven applications, where data circumvents, or at least alleviates, the need of imposing a closed form constitutive model. Similarly, chapter 3 treats the problem of data completion. A latent problem intrinsically associated to work with data is data scarcity. The truth is that nowadays testing facilities does not allow to obtain data points everywhere is the parametric domain but in a very limited region. As a consequence, numerical techniques able to infer (or complete) data outside the control points are of crucial interest. We are aware that the proposed methodologies constitute only a first step towards the final objective, letting plenty of room for new research lines.

Data-Driven Techniques

Abstract Standard simulation in classical mechanics is based on the use of two very different types of equations. The first one, of axiomatic character, is related to balance laws (momentum, mass, energy, ...), whereas the second one consists of models that scientists have extracted from collected, natural or synthetic data. Even if one can be confident on the first type of equations, the second one contains modeling errors. Moreover, this second type of equations remains too particular and often fails in describing new experimental results. Therefore, a new field in computer science called Data-Driven is gathering a lot of interest in the recent years. Its main objective is to strengthen the link between data and existing models, either by performing accurate parameter identification or by enriching existing models based on data.

Contents

The purpose of this chapter is to propose different methodologies to strengthen the link between experiments and simulation within the so-called Data-Driven framework. Section 2.1 presents a methodology that links existing data with numerical simulations in the solid mechanics framework. These simulations will employ universal balance laws while minimizing the need of employing explicit constitutive models in its closed form. This technique takes advantage of the use of manifold learning methodologies, that allow to extract the relevant information from large experimental datasets. Then, an iterative algorithm that finds the best candidate solution is constructed, i.e. the solution point lies inside the data set and it satisfies the conservation laws. Section 2.2 addresses the problem of data generation. Indeed, a numerical algorithm that identifies as much constitutive points as possible from heterogeneous experiments is proposed. Section 2.3 constitutes a step forward in order to define effective macroscopic constitutive manifolds from microscopic calculations. A non-Newtonian fluid acting on the micro scale engendrates a data set (or manifold) which allows to derive an effective Darcy's law at the macro scale level. The macroscopic mechanical problem is solved from the data contained in these constitutive manifolds without having to determine explicitly an homogenized or upscaled constitutive equation.

These three topics are addressed in three published papers that constitute this chapter:

- R. Ibáñez, E. Abisset-Chavanne, J. V. Aguado, D. Gonzalez, E. Cueto, F. Chinesta, *A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity.* Archives of Computational Methods in Engineering, **25**, 47-57, 2018.

- R. Ibáñez, D. Borzacchiello, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, P. Ladeveze, F. Chinesta, *Data-Driven non-linear elasticity: constitutive manifold construction and problem discretization.* Computational Mechanics, **60**, 813-826, 2017.

- R. Ibáñez, A. Scheuer, E. Lopez, E. Abisset-Chavanne, F. Chinesta, R. Keunings, *From elastic homogeneization to upscaling of non-Newtonian fluid flows in porous media.* International Journal of Material Forming, **1**, 1-11, 2017.

2.1 A Manifold Learning Approach to Data-Driven Computational Elasticity and Inelasticity

In the present work we will assume that all the needed experimental data is available. We will not address all the difficulties related to data generation or obtention from adequate experiments. This is a hot topic that will be addressed in section 2.2, but of course, remains the current investigation line of many research groups. On the contrary, we develop a method in which this stream of data plays the role of a constitutive equation, alleviating the need of a phenomenological fitting to a prescribed model.

To better understand the data-driven rationale, let us consider, for the sake of clarity, a very simple problem: linear elasticity. In that case the balance of (linear and angular) momentum leads to the existence of a symmetric second-order tensor $\boldsymbol{\sigma}$ (the so-called

Cauchy's stress tensor) verifying equilibrium, expressed in the absence of body forces and neglecting inertial terms, as

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0}.$$

The finite-element solution of this equilibrium equation starts from establishing a weak form in the domain Ω with boundary $\Gamma \equiv \partial\Omega$,

$$\int_{\Omega} \boldsymbol{u}^* \cdot (\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}) \ d\Omega = 0.$$

After integration by parts, it results

$$\int_{\Omega} \boldsymbol{\nabla}\boldsymbol{u}^* : \boldsymbol{\sigma} \ d\Omega = \int_{\Gamma} \boldsymbol{u}^* \cdot (\boldsymbol{\sigma} \cdot \boldsymbol{n}) \ d\Gamma,$$

where $\boldsymbol{n}$ represents the outward unit vector normal to the boundary.

If we consider $\Gamma = \Gamma_D \cup \Gamma_N$, $(\Gamma_D \cap \Gamma_N = \emptyset)$, representing portions of the domain boundary where, respectively, displacements $\boldsymbol{u} = \boldsymbol{u}_g(\boldsymbol{x})$ (Dirichlet boundary conditions) and tractions $\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{t}_g(\boldsymbol{x})$ (Neumann boundary conditions) are enforced, the weak form finally reads:

Find the displacement field $\boldsymbol{u} \in (\mathcal{H}^1(\Omega))^3$ satisfying the essential boundary conditions $\boldsymbol{u}(\boldsymbol{x} \in \Gamma_D) = \boldsymbol{u}_g(\boldsymbol{x})$ such that

$$\int_{\Omega} \boldsymbol{\varepsilon}^* : \boldsymbol{\sigma} \ d\Omega = \int_{\Gamma_N} \boldsymbol{u}^* \cdot \boldsymbol{t} \ d\Gamma, \tag{2.1}$$

$\forall \boldsymbol{u}^*$ *regular enough and vanishing on Γ_D, i.e. $\forall \boldsymbol{u}^* \in \left(\mathcal{H}_0^1(\Omega)\right)^3$.*

In the previous weak form, the symmetry of $\boldsymbol{\sigma}$ implies the equality $\nabla \boldsymbol{u} : \boldsymbol{\sigma} = \nabla^S \boldsymbol{u} : \boldsymbol{\sigma}$, with $\nabla^S \boldsymbol{u}$ the symmetric component of the displacement gradient, also known as strain tensor, generally denoted by $\boldsymbol{\varepsilon}$.

The weak form given by Eq. (2.1) involves kinematic and dynamic variables from the test displacement field $\boldsymbol{u}^*$ and the stress tensor $\boldsymbol{\sigma}$ respectively. In order to solve it a relationship linking kinematic and dynamic variables is required, the so-called constitutive equation. The simplest one, giving rise to linear elasticity, is known as Hooke's law (even if, more than a law, it is simply a model), and writes

$$\boldsymbol{\sigma} = \lambda \mathrm{Tr}(\boldsymbol{\varepsilon})\boldsymbol{I} + \mu\boldsymbol{\varepsilon}, \tag{2.2}$$

where $\mathrm{Tr}(\bullet)$ denotes the trace operator, and λ and μ are the Lame coefficients directly related to the Young modulus E and the Poisson's coefficient ν.

By introducing the constitutive model, Eq. (2.2), into the weak form of the momentum balance, Eq. (2.1), a problem is obtained that can be formulated entirely in terms of the displacement field $\boldsymbol{u}$. By discretizing it, using standard finite element approximations, for instance, and performing numerically the integrals involved in Eq. (2.1), we finally obtain a linear algebraic system of equations, from which the nodal displacements can be obtained.

In the case of linear elasticity there is no room for discussion: the approach is simple, efficient and has been applied successfully to many problems of interest. Today, there are numerous commercial codes making use of this mechanical behavior and nobody doubts about its pertinence in engineering practice. However, there are other material behaviors for whom simple models fail to describe any experimental finding. These models lack of

generality (universality) and for this reason a mechanical system is usually associated to different models that are progressively adapted and/or enriched from the collected data.

The biggest challenge could then be formulated as follows: can simulation proceed directly from data by circumventing the necessity of establishing a constitutive model? In the case of linear elasticity it is obvious that such an approach lacks of interest. However, in other branches of engineering science and technology it should be an appealing alternative to standard constitutive model-based simulations. In our opinion, we are at the beginning of a new era, the one of data-based or, more properly, data-driven engineering science and technology, where as much as possible data should be collected and information extracted in a systematic way by using adequate machine learning strategies. Then, simulations could proceed directly from this automatically acquired knowledge.

Thus, the question from a methodological viewpoint could be reformulated as: If Hooke had never existed, linear elasticity finite element simulations would have existed?

This section addresses this question, trying to push it beyond linear elastic behaviors. Subsection 2.1.1 focuses on the construction of the so-called constitutive manifold from the collected data, subsection 2.1.2 introduces data-driven simulation in the context of elastic models (linear and nonlinear). Finally, subsection 2.1.3 extends the procedure to inelastic behaviors.

2.1.1 Constructing the constitutive manifold

Imagine, to begin with (more general scenarios will soon be considered) mechanical tests conducted on a perfectly linear elastic material, in a specimen exhibiting uniform stresses and strains. As previously indicated, in this work we do not address issues related to data generation. Thus, for M randomly applied external loads, we assume ourselves able to collect M couples $(\boldsymbol{\sigma}_m, \boldsymbol{\varepsilon}_m)$, $m = 1, \ldots, M$. These pairs could be represented as a single point $\boldsymbol{X}_m$ in a phase space of dimension $D = 12$ (the six distinct components of the stress and strain tensors, respectively). In the sequel Voigt notion will be considered, i.e. stress and strain tensors will be represented as vectors and the fourth-order elastic tensor reduces to a square matrix.

Each vector $\boldsymbol{X}_m$ thus defines a point in a space of dimension D and, therefore, the whole set of samples represents a set of M points in $\mathbb{R}^D$. We conjecture that all these points belong to a certain low-dimensional manifold embedded in the high-dimensional space $\mathbb{R}^D$. Imagine for a while that the M points belong to a curve, a surface or a hyper-surface of dimension $d \ll D$. When $D = 3$ a simple observation suffices for checking if these points are located on a curve (one-dimensional manifold) or on a surface (two-dimensional manifold). However, when dealing with high dimensional spaces, a simple visual observation is, in general, not possible. Moreover, the extraction of uncorrelated features (often referred to as *latent* parameters) seems to be more physically pertinent.

Therefore, appropriate manifold learning (or non-linear dimensionality reduction) techniques are needed to extract the underlying manifold (when it exists) in multidimensional phase spaces. A panoply of techniques exist to this end. The interested reader can refer to [Tenenbaum *et al.* 2000, Roweis & Saul 2000, Polito & Perona 2001, Wang 2012, Amsallem & Farhat 2008], just to cite a few references. In this work we focus on the particular choice of Locally Linear Embedding —LLE— techniques [Roweis & Saul 2000]. This method proceeds in two steps and for the sake of completeness is detailed herein:

1. Each point $\boldsymbol{X}_m$, $m = 1, \ldots, M$ is linearly interpolated from its K nearest neighbors. In principle K should be greater that the expected dimension d of the underlying manifold and the neighbors should be close enough so as to ensure the validity of

linear approximation. In general, a small but enough number of neighbors K and a large-enough sampling M ensures a satisfactory reconstruction. For each point $\boldsymbol{X}_m$ we can write the locally linear data reconstruction as:

$$\boldsymbol{X}_m = \sum_{i \in \mathcal{S}_m} W_{mi} \boldsymbol{X}_i, \tag{2.3}$$

where W_{mi} are the unknown weights and $\mathcal{S}_m$ the set of the K-nearest neighbors of $\boldsymbol{X}_m$.

If we perform this locally linear interpolation for every data point in the high dimensional phase space, the set of weights that best approximates the manifold structure of the data will be obtained by minimizing the functional

$$\mathcal{F}(\boldsymbol{W}) = \sum_{m=1}^{M} \left\| \boldsymbol{X}_m - \sum_{i=1}^{M} W_{mi} \boldsymbol{X}_i \right\|^2,$$

where W_{mi} is zero if $\boldsymbol{X}_i$ does not belong to the set of K-nearest neighbors of $\boldsymbol{X}_m$.

2. We assume now that each linear patch around $\boldsymbol{X}_m$, $\forall m$, is mapped onto a lower dimensional embedding space of dimension $d \ll D$. To maintain the neighborhood structure of the set (other methods like isomap [Tenenbaum *et al.* 2000] conserve distance in the embedding space instead), weights are assumed to remain unchanged in the low-dimensional, embedding space. The problem thus becomes the determination of the coordinates of each point $\boldsymbol{X}_m$ in the low dimensional embedding space, $\boldsymbol{\xi}_m \in \mathbb{R}^d$.

For this purpose a new functional $\mathcal{G}$ is introduced, that depends on the searched coordinates $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M$

$$\mathcal{G}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M) = \sum_{m=1}^{M} \left\| \boldsymbol{\xi}_m - \sum_{i=1}^{M} W_{mi} \boldsymbol{\xi}_i \right\|^2,$$

where now the weights are known and the reduced coordinates $\boldsymbol{\xi}_m$ are unknown. The minimization of functional $\mathcal{G}$ results in a $M \times M$ eigenvalue problem whose d-bottom non-zero eigenvalues define the set of orthogonal coordinates in which the manifold is mapped.

It is important to note that functional $\mathcal{G}(\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_M)$, with the different coordinates $\boldsymbol{\xi}_m$ already calculated as just described, offers an error estimator on the locally linear embedding capacity, and even a local estimator can be derived by considering

$$\mathcal{E}(\boldsymbol{\xi}_m) = \left\| \boldsymbol{\xi}_m - \sum_{i=1}^{M} W_{mi} \boldsymbol{\xi}_i \right\|. \tag{2.4}$$

Thus, if we consider the introduction of a new point $\boldsymbol{\xi}$ in the embedding space $\mathbb{R}^d$ after identifying its neighbors set $\mathcal{S}(\boldsymbol{\xi})$ and calculating the locally linear approximation weights, we can come back to $\mathbb{R}^D$ and reconstruct $\boldsymbol{X}$ from its neighbors $\boldsymbol{X}_i$, $i \in \mathcal{S}(\boldsymbol{\xi})$.

In the linear elastic behavior the application of the just described technique results, as expected, in a flat manifold of dimension two, i.e. $d = 2$. This is in perfect agreement to the fact that Hooke's law is completely characterized by two coefficients (either Young's modulus and Poisson coefficient, or Lame's coefficients) and is linear. Fig. 2.1 depicts the location of samples $\boldsymbol{\xi}_m = \boldsymbol{\xi}(\mathbf{X}_m) = \boldsymbol{\xi}(\boldsymbol{\sigma}_m, \boldsymbol{\varepsilon}_m)$ into the resulting two-dimensional manifold,
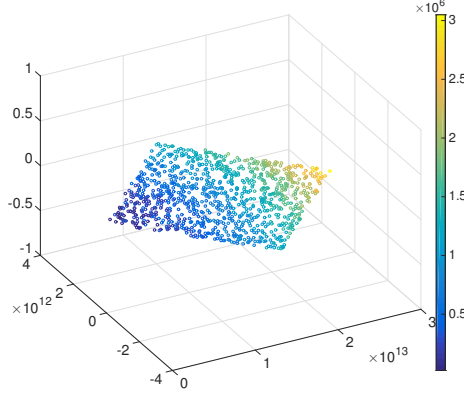
Figure 2.1 – Reduced coordinates $\boldsymbol{\xi}_m$ on the resulting two-dimensional manifold. The color map represents the associated elastic energy.

as well as the associated elastic energy of each sample, showing that LLE preserves the smoothness of the elastic energy field of the sample in the embedding space.

We have abandoned the idea of a phenomenological constitutive equation. Instead, we have defined the concept of (experimentally obtained) constitutive manifold, as the one with a minimal number of latent parameters (embedding coordinates) in which the state of the sample will evolve in different stress and strain conditions.

However, for the method to be useful, we need to define a strategy to solve problems stated in weak form and discretized by finite elements. Several options can be considered, which are described next.

1. **Identifying the locally linear behavior**. If we consider locally linear approximations, fully justified if $\mathcal{E}(\boldsymbol{\xi}_m)$, given by Eq. (2.4), remains small enough at each position $\boldsymbol{\xi}_m$ (if it is not the case the sampling should be improved locally or globally), we can write

$$\boldsymbol{\xi}_m = \sum_{i=1}^{M} W_{mi} \boldsymbol{\xi}_i,$$

with $W_{mi} = 0$ if $i \notin \mathcal{S}_m$ and where $\boldsymbol{\xi}_m$ is a stress-strain couple. This implies a locally linear elastic behavior, that allows obtaining the elastic tensor $\mathbf{C}$ from $\boldsymbol{X}_m$ and $\boldsymbol{X}_i$ (related to $\boldsymbol{\xi}_m$ and $\boldsymbol{\xi}_i$ respectively), with $i \in \mathcal{S}_m$, by minimizing the functional

$$\mathcal{H}(\mathbf{C}) = \sum_{i \in \mathcal{S}_m} (\boldsymbol{\sigma}_i - \mathbf{C} \cdot \boldsymbol{\varepsilon}_i)^2.$$

This results in the obtention of $\mathbf{C}(\boldsymbol{X}_m) \equiv \mathbf{C}_m$.

2. **Identifying the locally linear *tangent* behavior**. In order to consider Newton strategies the locally tangent linear behavior should be computed. Again, it is easy to obtain by considering $\Delta_{mi} \equiv \boldsymbol{X}_m - \boldsymbol{X}_i = (\boldsymbol{\sigma}_m - \boldsymbol{\sigma}_i, \boldsymbol{\varepsilon}_m - \boldsymbol{\varepsilon}_i)$ or $\Delta_{mi} = (\Delta\boldsymbol{\sigma}_i^m, \Delta\boldsymbol{\varepsilon}_i^m)$, $i \in \mathcal{S}_m$. Because of the locally linear behavior around point $\boldsymbol{X}_m$, we can write

$$\Delta\boldsymbol{\sigma}_i^m = \mathbf{C}_T \cdot \Delta\boldsymbol{\varepsilon}_i^m, \tag{2.5}$$

that allows defining the functional $\mathcal{H}_T(\mathbf{C}_T)$

$$\mathcal{H}_T(\mathbf{C}_T) = \sum_{i \in \mathcal{S}_m} (\Delta \boldsymbol{\sigma}_i^m - \mathbf{C}_T \cdot \Delta \boldsymbol{\varepsilon}_i^m)^2, \qquad (2.6)$$

whose minimization results in the tangent elastic tensor $\mathbf{C}_T(\boldsymbol{X}_m) \equiv \mathbf{C}_{T,m}$.

3. **No identification of the local behaviour**. The third level of description considers points $\boldsymbol{X}_m$ without trying to identify local behavior models at all.

It is important to note that even if the just discussed descriptions are based on the original manifold $\boldsymbol{X}_m$ and not on the reduced one $\boldsymbol{\xi}_m$, the consideration of the reduced manifold allows to obtain a global view of the manifold dimensionality as well as faster interpolations on the manifold.

2.1.2 Data-driven simulation in the elastic case

We assume that the elastic behavior is accessible from the data contained into the so-called constitutive manifold but that an explicit expression relating stresses and strains is neither available nor desired. Immediately, a question arises on how to solve the weak form related to the equilibrium of the mechanical system given by Eq. (2.1) if no closed-form expression on $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon})$ is available.

In this case we could consider three different approaches depending on the chosen behavior description as just discussed in subsection (2.1.1):

1. From the just identified locally linear behavior $\mathbf{C}(\boldsymbol{X})$ one could apply the simplest explicit linearization technique operating on the standard weak form

$$\int_\Omega \boldsymbol{\varepsilon}^*(\boldsymbol{x}) : \boldsymbol{\sigma}^{n+1}(\boldsymbol{x}) \, d\Omega = \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\Omega, \qquad (2.7)$$

where at each point, from the stress-strain couple at position $\boldsymbol{x}$, $\boldsymbol{X}(\boldsymbol{x})$, the locally linear behavior $\mathbf{C}(\boldsymbol{X}(\boldsymbol{x}))$ can be obtained (in practice at the Gauss points used for the integration of the weak form) that allows us to write (using Voigt notation)

$$\int_\Omega \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot \left(\mathbf{C}(\boldsymbol{X}^n) \cdot \boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}) \right) \, d\Omega = \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\Gamma.$$

This allows, in turn, to compute the displacement field and from it, to update the strain and stress fields, to compute again the locally linear behavior. The process continues until convergence.

2. From the just identified locally linear *tangent* behavior $\mathbf{C}_T(\boldsymbol{X})$ one could apply a Newton linearization technique where

$$\boldsymbol{\sigma}^{n+1}(\boldsymbol{\varepsilon}^n + \Delta \boldsymbol{\varepsilon}) = \boldsymbol{\sigma}^n(\boldsymbol{\varepsilon}^n) + \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{\varepsilon}} \Delta \boldsymbol{\varepsilon} = \boldsymbol{\sigma}^n(\boldsymbol{\varepsilon}^n) + \mathbf{C}_T \cdot \Delta \boldsymbol{\varepsilon},$$

that, once introduced into the weak form, reads

$$\int_\Omega \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot (\mathbf{C}_T(\boldsymbol{X}^n) \cdot \Delta \boldsymbol{\varepsilon}(\boldsymbol{x})) \, d\Omega$$

$$= -\int_\Omega \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot (\mathbf{C}(\boldsymbol{X}^n) \cdot \boldsymbol{\varepsilon}^n(\boldsymbol{x})) \, d\Omega + \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\Gamma.$$
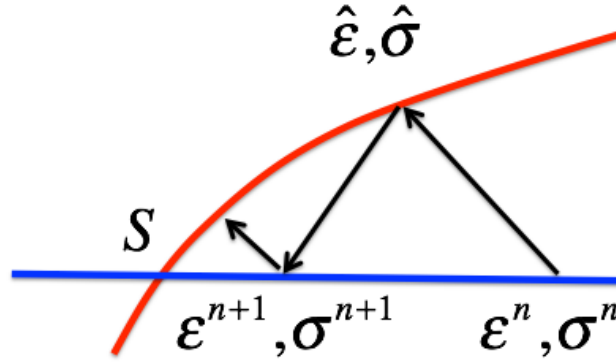
Figure 2.2 – A generic nonlinear iteration solver between the constitutive manifold (red) and the equilibrium manifold (blue), representing the locus of the points satisfying the weak form of the problem in mixed form, Eq. (2.7).

3. If no local behavior has been identified, the only knowledge consists of the experimental data. In these circumstances we propose to consider a mixed formulation involving the two unknown fields $\varepsilon(\boldsymbol{u})$ and $\boldsymbol{\sigma}$ as considered in the LaTIn method [Ladeveze 1989]. We consider a simple solution strategy consisting on an iteration between two manifolds, the first one related to $(\varepsilon, \boldsymbol{\sigma})$ couples verifying equilibrium Eq. (2.1); and the second one related to couples $(\hat{\varepsilon}, \hat{\boldsymbol{\sigma}})$ verifying the (unknown) constitutive equation —in other words, belonging to the constitutive manifold. The iteration solver sketched in Fig. 2.2, depicts the usually non linear constitutive manifold (red curve) and the equilibrium one (in blue). The problem solution is found at the intersection of both manifolds.

If we assume that, at iteration n, the couple $(\varepsilon^n, \boldsymbol{\sigma}^n)$ verifies the equilibrium, and that it does not belong to the constitutive manifold, a new couple $(\hat{\varepsilon}, \hat{\boldsymbol{\sigma}})$ is sought by considering an appropriate search direction from $(\varepsilon^n, \boldsymbol{\sigma}^n)$. In fact the searched couple is no more that the intersection of the search direction with the constitutive manifold. The just updated stress-strain couple belongs to the constitutive manifold, but it does not verify equilibrium. Thus, a new equilibrated solution $(\varepsilon^{n+1}, \boldsymbol{\sigma}^{n+1})$ is searched from the former one, being the intersection of a new search direction and the equilibrium manifold. The iteration process continues until reaching the problem solution at the intersection of both manifolds.

The just described procedure requires a local step for the computation of the couple $(\hat{\varepsilon}, \hat{\boldsymbol{\sigma}})$ at each integration point considered in the weak form, Eq.(2.1), and a global step in which the weak form is solved with the behavior known at all the integration points. In what follows we describe both steps.

- Local step

  At each integration point $\boldsymbol{x}_g$, $g = 1, \ldots, \texttt{ngp}$, we consider $(\varepsilon^n(\boldsymbol{x}_g), \boldsymbol{\sigma}^n(\boldsymbol{x}_g))$ and look for $(\hat{\varepsilon}(\boldsymbol{x}_g), \hat{\boldsymbol{\sigma}}(\boldsymbol{x}_g))$. Even if there is an infinity of possible search directions, a natural choice consists in projecting it onto the constitutive manifold.

- Global step

  From the strain-stress couples satisfying the constitutive law at every integration point, we come back to the weak form, Eq. (2.1), in order to obtain updated

strain-stress couples satisfying equilibrium $(\boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}), \boldsymbol{\sigma}^{n+1}(\boldsymbol{x}))$, $\boldsymbol{x} \in \Omega$.

The generic search direction can be written as:

$$\boldsymbol{\sigma}^{n+1}(\boldsymbol{x}) - \hat{\boldsymbol{\sigma}}(\boldsymbol{x}) = \mathbf{D} \cdot (\boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}) - \hat{\varepsilon}(\boldsymbol{x})), \tag{2.8}$$

with $\mathbf{D}$ a symmetric positive-definite matrix to ensure the problem ellipticity discussed below. Enforcing now the equilibrium

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot \boldsymbol{\sigma}^{n+1}(\boldsymbol{x}) \, d\boldsymbol{x} = \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\boldsymbol{x},$$

and using Eq. (2.8), it results

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot \left(\hat{\boldsymbol{\sigma}}(\boldsymbol{x}) + \mathbf{D} \cdot (\boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}) - \hat{\varepsilon}(\boldsymbol{x})) \right) \, d\boldsymbol{x} = \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\boldsymbol{x},$$

that can be rewritten as

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot \left(\mathbf{D} \cdot \boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}) \right) d\boldsymbol{x} =$$
$$- \int_{\Omega} \boldsymbol{\varepsilon}^*(\boldsymbol{x}) \cdot (\hat{\boldsymbol{\sigma}}(\boldsymbol{x}) - \mathbf{D} \cdot \hat{\varepsilon}(\boldsymbol{x})) \, d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{u}^*(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) \, d\boldsymbol{x}. \tag{2.9}$$

Matrix $\mathbf{D}$ should provide the fastest convergence rate while ensuring the problem ellipticity. To ensure its positivity we can consider $\mathbf{D} = \boldsymbol{B}^2$ with $\boldsymbol{B}$ symmetric, i.e. $\boldsymbol{B}^T = \boldsymbol{B}$, and look for $\boldsymbol{B}$ instead of $\mathbf{D}$.

The *a priori* choice of direction $\mathbf{D}$ is not obvious in most of problems. In the case of the LaTIn method [Ladeveze 1989] this matrix is assumed given when solving the global problems precisely because it was proposed as a nonlinear solver able to decouple the local and nonlinear problem from the global but linear one. In our case, we are considering a mixed formulation for solving a problem without an explicit knowledge of the constitutive equation. The most general option consists on considering matrix $\mathbf{D}$ unknown. Thus, our strategy is composed of a sequence of nonlinear-local and nonlinear-global problems, trying to avoid *a priori* choices of $\mathbf{D}$. Obviously if the last is fixed, global problems become linear as it is the case when considering the LaTIn linearization technique. Moreover, the discrete global matrix does not change during the iterations. However, we would like to emphasize that our objective is to solve a constitutive model-free problem, more than addressing nonlinear issues.

Thus, we distinguish two type of iterations, the so-called global-local ones that involves the determination of stress-strain couples verifying the constitutive equation and then their updating to ensure equilibrium (as illustrated in Fig. 2.2). Then a second iteration is needed for solving the nonlinear global problem in order to compute the stress-strain couple verifying equilibrium when the searching direction $\mathbf{D}$ is assumed unknown. This induces an additional nonlinearity in the global equilibrium problem.

At this point two possibilities exist:

(a) Considering a single direction $\mathbf{D}$, the same for every Gauss point for which the behavior was determined. Each of them is represented by a point on the constitutive manifold. In that case in order to determine the stress-strain couple satisfying equilibrium as well as the optimal direction $\mathbf{D}$,

we are enforcing Eq. (2.9) as well as the fact that the searched couple $(\boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}), \boldsymbol{\sigma}^{n+1}(\boldsymbol{x}))$ must be the closest point to the constitutive manifold. This optimality condition writes

$$\mathbf{D} = \operatorname{argmin}_{\mathbf{D}^*} \left(\left(\boldsymbol{\sigma}^{n+1}(\boldsymbol{x}; \mathbf{D}^*) - \hat{\boldsymbol{\sigma}}^* \right)^2 + \left(\boldsymbol{\varepsilon}^{n+1}(\boldsymbol{x}; \mathbf{D}^*) - \hat{\boldsymbol{\varepsilon}}^* \right)^2 \right), \quad (2.10)$$

where $(\hat{\boldsymbol{\sigma}}^*, \hat{\boldsymbol{\varepsilon}}^*)$ is the closest point on the constitutive manifold to the stress-strain couple related to the direction $\mathbf{D}^*$.

Obviously the solution requires some iterations to reach the minimum distance that will be in general (except when considering linear behaviors) non-zero because we consider the same matrix $\mathbf{D}$ for all the Gauss points involved in the integration of the weak form (2.9).

(b) We consider a field $\mathbf{D}(\boldsymbol{x})$, that implies the increase of the number of degrees of freedom. However, by considering for example a different matrix at each Gauss point, the minimization problem given by Eq. (2.10) leads to the problem solution in a single iteration. The employ of a coarse mesh to approximate $\mathbf{D}$ is a nice compromise between the two limit cases: considering a single search direction or one at each Gauss point.

2.1.2.1   A first numerical example: A beam subjected to simple traction

In order to illustrate the data-driven procedure, we consider first a linear elastic beam subjected to simple traction and solve the associated 1D equilibrium problem. Different scenarios are considered and discussed below.

First, the beam is assumed clamped at its left boundary $x = 0$ with a constant unit traction force $F = 1$ applied at its right boundary $x = 1$. Because of the expected simple solution only 5 linear finite elements were considered for discretizing its equilibrium weak form. Fig. 2.3 depicts the constitutive manifold. In a general setting, this manifold should come from experiments, but in this case was generated *in silico* by assuming a linear elastic behavior with an unit elastic modulus.

The use of strategies based on the identification of the locally linear behavior or its tangent counterpart allows as expected (due to its linear behavior) solving the problem in a single iteration. It is important to note that both strategies are weakly intrusive, making possible its implementation into any commercial simulation code with the only difference that the updated locally linear behavior comes form a data table instead of any mathematical expression.

In what follows we are discussing the use of the third strategy. The equilibrium manifold and the different strain-stress couples at the different iterations are depicted in Fig. 2.3 for $D = 10$, $D = 2$ and $D = 1$. These D-values represent in fact different search directions in Fig. 2.2. It can be noticed that when $D = 1$ is chosen, this value coincides with the elastic modulus associated to the constitutive manifold, and therefore convergence is reached in a single iteration. All the simulations started by assuming the same stress-strain couple $(\sigma^0, \varepsilon^0) = (3.0, 3.0)$ at every Gauss point.

In these figures, the search direction in the global problem D was fixed "a priori". When the strategy described in the previous section is used, implying the determination of the optimal value of D, the nonlinear problem involving σ, ε and D, with $(\sigma^0 = 3, \varepsilon^0 = 3, D^0 = 3)$, converges in a single iteration of the local-global problem. This is so even if a few iterations were required for solving the nonlinear global problem, to obtain the reference values defining the problem solution $(\sigma = 1.0, \varepsilon = 1.0, D = 1)$. Because of the linearity of the constitutive manifold, no difference exists between considering a single direction D or
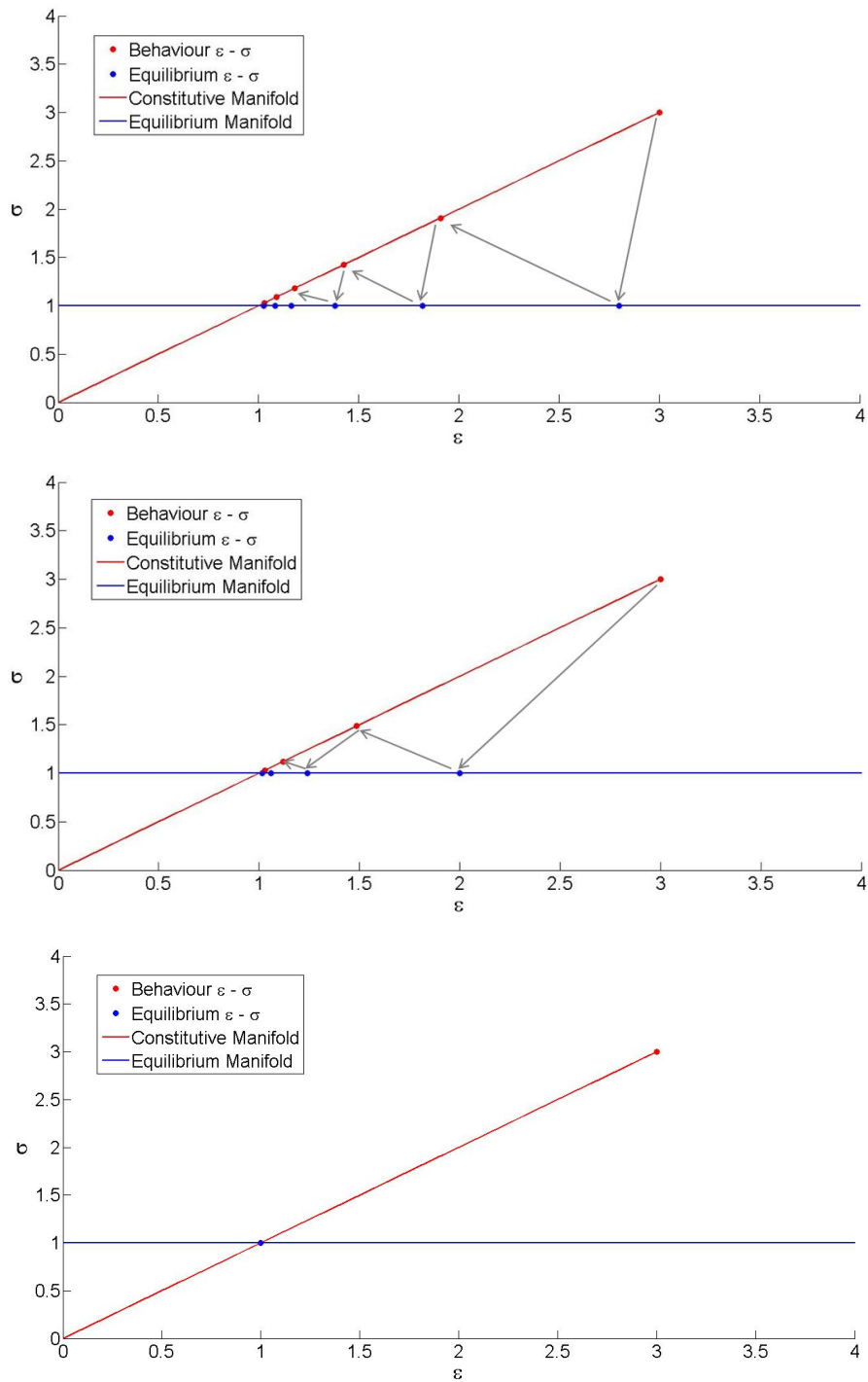
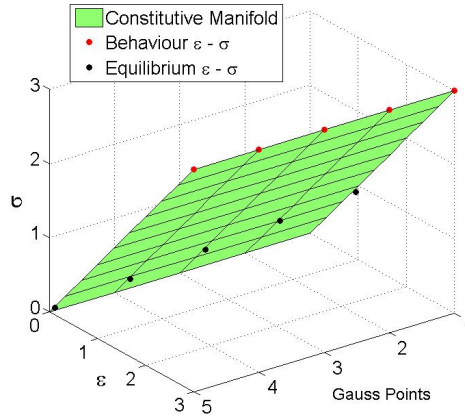Figure 2.3 – Beam subjected to traction: (top) $D = 10$, (center) $D = 2$ and (bottom) $D = 1$.

Figure 2.4 – Beam subjected to uniformly distributed traction.

a different one at each Gauss point. The solution is again obtained in a single global-local iteration and a few ones for solving the nonlinear global problem.

In order to make the problem a bit more complex, we consider the previous one but now we consider an uniformly distributed traction along the beam length. Thus a linear stress and strain distribution is expected. In other words, each Gauss point will be at a state located at different points of the constitutive manifold. Fig. 2.4 represents the stress-strain manifold along the beam length, where the stress-strain couples at the Gauss points are shown. It can be seen that when starting from the initial guess $(\sigma^0(x) = 3, \varepsilon^0(x) = 3, D^0 = 3)$ and again because of the linearity of the constitutive manifold, the convergence is reached in a single global-local iteration with few iterations for the solution of the nonlinear global problem.

Finally, we consider a nonlinear constitutive law defined from points with a prescribed stress-strain relationship $\sigma = E\epsilon^2$, with $E = 1$. In the case of a unit traction at the right boundary and when considering uniform initial strain and stress guesses on the constitutive manifold, all the Gauss points will have an identical behavior.

When applying the fixed point linearization based on the locally linear manifold **C** or the Newton strategy making use of the locally linear tangent manifold $\mathbf{C}_T$, the procedure proposed in the previous section converges very fast. Iterations to convergence are depicted in Figures 2.5 and 2.6 respectively.

If, on the contrary, we proceed following the third strategy mentioned previously, i.e., directly from data, Figure 2.7 depicts the initial guess and the solution after convergence $(\sigma(x) = 1, \epsilon(x) = 1)$. Here, D is unique and calculated at each global-local iteration. Moreover, at each one of these iterations a nonlinear global problem must be solved needing for few extra-iterations.

If we combine behavior nonlinearities and nonuniform solutions (e.g., a distributed traction along the bar) we proved that the convergence can be improved by considering a different D at each Gauss point with respect to the use of a single search direction D for all them, even if the global problem size increases significantly.

Manifold-based locally linear behaviors resulting in the fixed point and Newton strategies proceed faster that the one based on the solution from the only knowledge of data. However, it requires the identification of such behaviors with the subsequent errors that
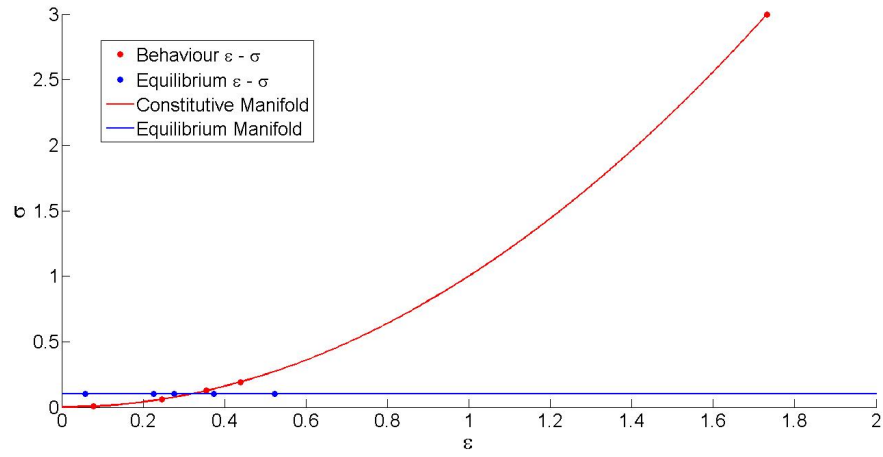
Figure 2.5 – Beam subjected to a traction for a nonlinear behavior: manifold-based fixed point linearization
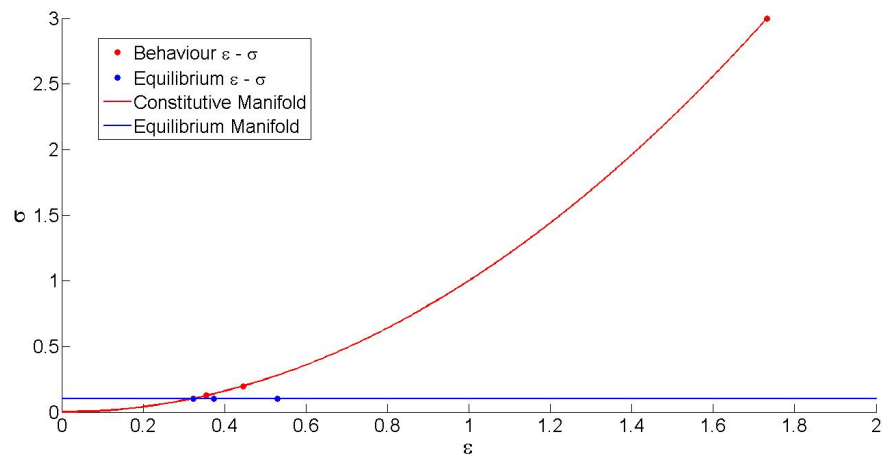

Figure 2.6 – Beam subjected to a traction for a nonlinear behavior: manifold-based Newton linearization
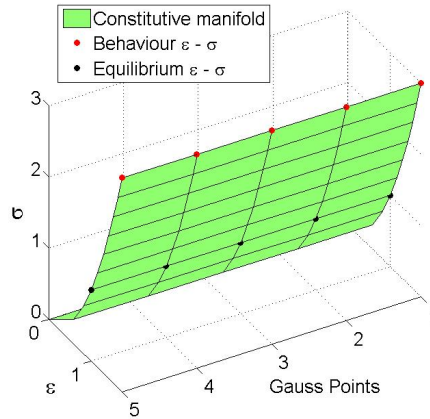
Figure 2.7 – Beam subjected to traction for a nonlinear behavior.

they could imply if coarse samplings of the constitutive behavior are employed.

2.1.2.2 A two-dimensional case study

We considered a 2D problem defined on a square involving again an elastic behavior defined from a manifold in the space $(\boldsymbol{\sigma}, \boldsymbol{\varepsilon})$. This constitutive manifold proved to project onto a just two-dimensional one in its reduced form, as discussed previously.

The square is clamped on its left boundary, free on the top and bottom sides and a unit traction is applied on its right side. Any of the proposed strategies, the ones making use of the manifold-based locally linear behaviors or the one proceeding directly from data, allow reaching the same converged solution depicted in Fig. 2.8. The last one employs a single search direction $\mathbf{D}$ or a different one at each Gauss point $\mathbf{D}(\boldsymbol{x})$. It agrees in minute with the one obtained by using standard model-based discretization. Again, a Newton technique remains superior to the other choices.

In what respects the solution accuracy there are different aspects affecting it: (i) the constitutive manifold sampling when nonlinear behaviors are addressed; (ii) the finite element approximation and finally (iii) the threshold consider in the nonlinear iteration schemes. Even if a detailed analysis of the accuracy and rate of convergence is beyond the aim of the present work, our numerical experiments indicate that convergence is assured by using fine enough samplings of the constitutive manifolds as well as by considering fine enough finite element discretizations.

2.1.3 Addressing inelastic behaviors: Linear elastic-perfectly plastic behavior

In this section we start by addressing the case of a linear-elastic-perfectly plastic 2D behavior. We assume the linear elastic contribution defined locally from $\mathbf{C}(\boldsymbol{X}^e)$ ($\boldsymbol{X}^e$ refers to the stress-elastic strain manifold) whereas the plastic contribution that involves the yield surface $f(\boldsymbol{\sigma})$ is assumed given by its own manifold.

Using again Voigt notation, the elastic behavior expressed from $\boldsymbol{\sigma} = \mathbf{C} \cdot \boldsymbol{\varepsilon}^e$, where $\mathbf{C}$ represents the manifold-based elastic tensor and $\boldsymbol{\varepsilon}^e$ refers to the elastic component of the
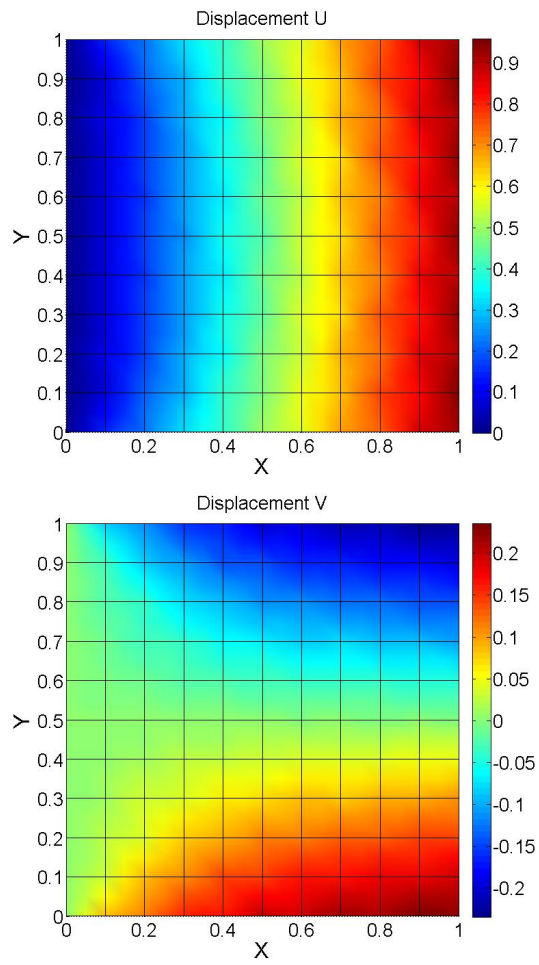
Figure 2.8 – 2D problem associated to a "hidden" linear elastic behavior: (top) horizontal component of the displacement and (bottom) vertical component
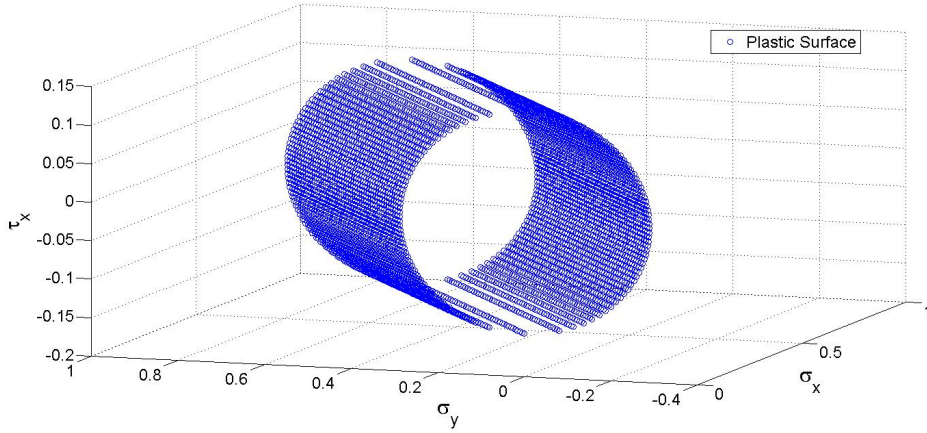
Figure 2.9 – Plastic manifold associated to the von Mises plasticity case.

deformation (the reversible one). The total strain can be decomposed in its elastic and inelastic components,

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p,$$

where we assume the plastic flow rate

$$\dot{\boldsymbol{\varepsilon}}^p = \lambda \frac{\partial f(\boldsymbol{\sigma})}{\partial \boldsymbol{\sigma}} = \lambda \boldsymbol{n},$$

where the yield surface $f(\boldsymbol{\sigma})$ is provided by experimental data. To generate these data *in silico*, we assume that it follows a von Mises model $f(\boldsymbol{\sigma}) = \sigma_e - Y$, with Y the yield stress (no hardening is considered) and σ_e the equivalent stress related to the von Mises criterion. $f(\boldsymbol{\sigma})$ results in the surface represented in Fig. 2.9 where, for the sake of clarity, it is represented in the space of stresses.

The persistency condition $\dot{f}(\boldsymbol{\sigma}) = 0$ when plastic flow occurs, results in the following plastic flow

$$\lambda = \frac{\boldsymbol{n}^T \cdot \mathbf{C} \cdot \dot{\boldsymbol{\varepsilon}}}{\boldsymbol{n}^T \cdot \mathbf{C} \cdot \boldsymbol{n}},$$

or in its incremental counterpart

$$\lambda = \frac{\boldsymbol{n}^T \cdot \mathbf{C} \cdot \Delta\boldsymbol{\varepsilon}}{\boldsymbol{n}^T \cdot \mathbf{C} \cdot \boldsymbol{n}},$$

with now $\Delta\boldsymbol{\varepsilon}^p = \lambda\boldsymbol{n}$.

Here three fields must be considered, stress, strain and plastic strain. As soon as the last one is known, the elastic strain can be locally determined and the stresses obtained from the elastic manifold using the couple stress-elastic component of the strain.

In these expressions everything is properly defined except $\boldsymbol{n}$, since we assume that the explicit form of the yield condition, i.e. $f(\boldsymbol{\sigma})$ is unknown and the only available data is the manifold depicted in Fig. 2.9. However, $\boldsymbol{n}$ is easily accessible by considering the normal vector to the plastic manifold depicted in Fig. 2.9.

Now one could imagine performing a standard linear elastic-perfectly plastic simulation by using a finite element explicit code where the plastic deformation is computed from the
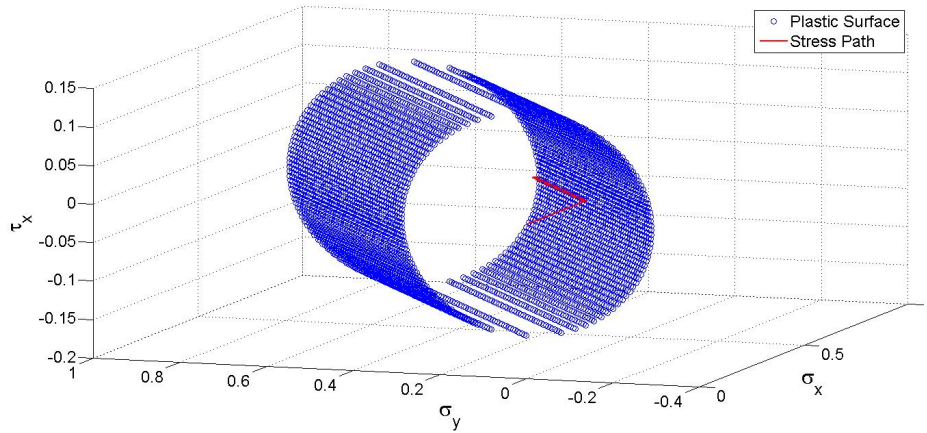
Figure 2.10 – Stress trajectory in the stress space in the elastic-perfecly plastic behavior

manifold that allows extracting $\boldsymbol{n}$ instead of the knowledge of function $f(\boldsymbol{\sigma})$ and its explicit derivative with respect to the stresses.

When considering the traction of a square domain along its right side, with appropriate boundary conditions on its left side (with tension-free conditions on the top and bottom boundaries) ensuring an homogeneous stress and strain fields everywhere in the domain, the stress trajectory in the stress space is depicted in Fig. 2.10. It can be noticed that the elastic behavior applies when the stress remains inside the plastic surface and then it remains in the surface during the plastic flow. Again, for the sake of simplicity, the results are shown in the stress domain. Finally, Fig. 2.11 depicts the three composites of the plastic strain for three different levels of the applied load acting on the right side of the clamped square previously considered. The different strategies allows to compute the same results.The Newton algorithm results again to be the one involving less computational effort.

Even if this analysis proved that we could proceed as usually when function $f(\boldsymbol{\sigma})$ is not explicitly known, the elastic behavior was assumed given by the locally-linear elastic manifold. Obviously the extension to implicit formulations or to more complex nonlinear elastic behaviors again based on a locally-linear tangent description is straightforward.

2.1.4 Conclusions

This work constitutes a first attempt to reduce the modeling needs in computational mechanics. We proved that by knowing the different stress-strain couples defining the elastic behavior as well as the manifold defining the yield condition there is no need to create models for representing neither the linear or nonlinear elastic behaviors nor the yield condition. Different linearization strategies have been proposed. Two of them are weakly intrusive and easily implantable in existing commercial simulation codes, since they are based on a locally-linear elastic expression. Another linearization strategy proceeding exclusively from data iterates from a local-nonlinear problem to ensure the verification of the constitutive behavior and a linear or nonlinear-global problem for ensuring the mechanical equilibrium.

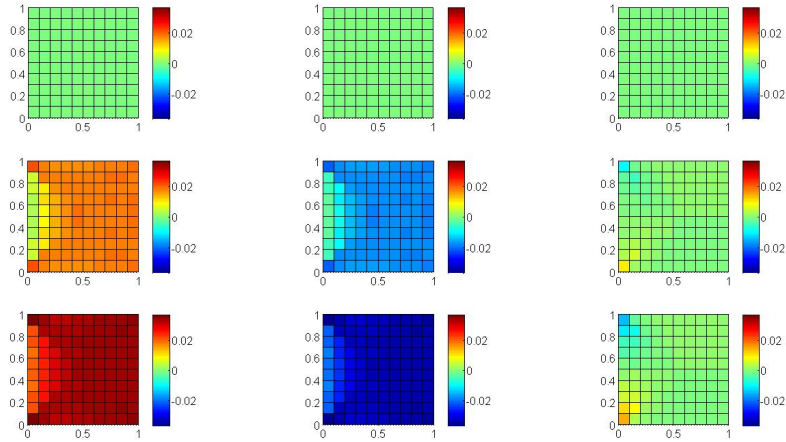Despite the fact of addressing quite simple problems, a great potential can be noticed,

Figure 2.11 – Plastic strain at the initial time (top), for the half of the total load (middle) and for the entire load (bottom), for components ϵ_{xx}^p (left), ϵ_{yy}^p (center) and ϵ_{xy}^p (right)

that could constitute a new paradigm in computational mechanics, linking experimental data with discretization techniques while reducing as much as possible the needs of modeling issues.

2.2 Data-Driven non-linear elasticity: constitutive manifold construction and problem discretization

Even though in section 2.1 we proposed some methods for performing simulation employing the just developed concept of constitutive manifolds arising from data. One of the current bottlenecks of Data-Driven based simulation is the large amount of required data, some of them inaccessible from the nowadays testing facilities. Indeed, a vast majority of current constitutive calibration procedures rely on relatively simple mechanical tests. The reader may think of simple traction, compression, shear or biaxial loadings. Moreover, standard calibration procedures strongly relies on the hypothesis that the stress/strain state throughout the material is homogeneous, or equivalently, only one point per experiment in this stress/strain space is activated. Thus, constructing the constitutive manifold by carrying out a sequence of homogeneous tests with the purpose of activating all the possible strain states, seems today too expensive, but probably not in the future where data is expected playing a major role. Such difficulty can be circumvented in many cases, and in any case alleviated, by considering complex tests, collecting as many data as possible and then using a data-driven inverse approach in order to infer the whole constitutive manifold from few but complex experimental tests, as discussed in the present work.

This section is organized as follows: subsection 2.2.1 presents a methodology to perform inverse identification, first in the linear setting and secondly in the non-linear case. Subsection 2.2.2 provides numerical results associated to the methodology presented in the previous subsection.

2.2.1 Constitutive manifold Identification

The inverse technique is introduced within the linear elasticity framework. Afterwards, all concept and notations introduced for the linear setting will be useful to extend the methodology to the non-linear framework. More in detail, the non-linear setting relies on constructing a polynomial approximation of the elastic energy, whose second derivative results in the elastic tensor, and whose identification from collected data seems to be more robust.

2.2.1.1 Linear setting

As mentioned before, our starting point of the inverse identification technique is linear elasticity. Hence, the elastic constitutive tensor no longer depends on the strain state i.e. $\mathbf{C}(\varepsilon) \to \mathbf{C}$. Eq. (2.11) shows the standard weak form, in absence of body forces and inertial terms:

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\mathbf{x}) : \boldsymbol{\sigma}(\mathbf{x}) \; d\Omega = \int_{\Gamma_N} \mathbf{u}^*(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) \; d\Gamma, \tag{2.11}$$

that using Voigt notation and the behavior derived from the constitutive manifold, becomes

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\mathbf{x}) \cdot \mathbf{C}\varepsilon(\mathbf{x}) \; d\Omega = \int_{\Gamma_N} \mathbf{u}^*(\mathbf{x}) \cdot \mathbf{t}(\mathbf{x}) \; d\Gamma. \tag{2.12}$$

By using an appropriate linearization, this last expression allows one to compute (at convergence) every mechanical field. However, as previously argued, prior to proceed with the calculations summarized above and analyzed in detail in 2.1, one must accomplish the construction of the so-called constitutive manifold.

Using a c-term parametrization of 6×6 matrices (the more general consisting of canonical matrices fulfilling symmetry constraints) we can write

$$\mathbf{C} = \sum_{i=1}^{c} \alpha_i \mathbf{M}_i, \tag{2.13}$$

with coefficients α_i unknown.

By introducing this matrix parameterization into the equilibrium weak form it results

$$\int_{\Omega} \boldsymbol{\varepsilon}^*(\mathbf{x}) \cdot \left(\left(\sum_{i=1}^{c} \alpha_i \mathbf{M}_i \right) \varepsilon(\mathbf{x}) \right) \; d\Omega = \int_{\Gamma_N} \mathbf{u}^*(\mathbf{x}) \cdot \mathbf{t} \; d\Gamma, \tag{2.14}$$

whose discrete form reads

$$\mathbf{U}^* \cdot \left(\sum_{i=1}^{c} \alpha_i \mathbf{K}_i \right) \mathbf{U} = \mathbf{U}^* \cdot \mathbf{T}, \tag{2.15}$$

with $\mathbf{K}_i$ the stiffness matrices corresponding to the $\mathbf{M}_i$ basis and $\mathbf{U}$ the nodal displacement vector.

We assume that local displacements, are accessible (experimentally measurable) at a certain region of the domain (in general a portion of its boundary). Their associated degrees of freedom are hereafter referred to with the superscript $\bullet^{\mathcal{O}}$. Thus, making use of

a partition of the displacement vector $\mathbf{U}^{\mathcal{O}}$ and $\mathbf{U}^{\mathcal{H}}$ referring to the observable and hidden displacements, respectively, the previous discrete system reads

$$\left(\begin{array}{cc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{O}} \end{array} \right) \left(\begin{array}{c} \mathbf{U}^{\mathcal{H}} \\ \mathbf{U}^{\mathcal{O}} \end{array} \right) = \left(\begin{array}{c} \mathbf{T}^{\mathcal{H}} \\ \mathbf{T}^{\mathcal{O}} \end{array} \right). \tag{2.16}$$

This system of equations is obviously complemented with appropriate Dirichlet boundary conditions on Γ_D. In the previous algebraic system, vectors $\mathbf{T}^{\mathcal{O}}$ and $\mathbf{T}^{\mathcal{H}}$ refer to the nodal traction contributions at nodes related to the observable and hidden displacements, respectively.

The algebraic system (2.16) has as unknowns the hidden displacements $\mathbf{U}^{\mathcal{H}}$ and the constitutive coefficients α_i, being known the observable displacements $\mathbf{U}^{\mathcal{O}}$. If the number of known displacements that corresponds with the size of vector $\mathbf{U}^{\mathcal{O}}$ is large enough (in all cases larger than the number of alpha-coefficients, c) it is thus possible to solve the resulting nonlinear algebraic problem to compute both the unknown displacements $\mathbf{U}^{\mathcal{H}}$ and the coefficients defining the material behavior α_i. In the opposite case it is always possible to apply some regularization to solve the undetermined resulting problem (e.g. Tikhonov regularization). In the sequel we focus in the former scenario.

System (2.16) can be rewritten as follows

$$\left(\begin{array}{cccc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \mathbf{K}_1^{\mathcal{H}\mathcal{O}} \mathbf{U}^{\mathcal{O}} & \cdots & \mathbf{K}_{\mathsf{c}}^{\mathcal{H}\mathcal{O}} \mathbf{U}^{\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \mathbf{K}_1^{\mathcal{O}\mathcal{O}} \mathbf{U}^{\mathcal{O}} & \cdots & \mathbf{K}_{\mathsf{c}}^{\mathcal{O}\mathcal{O}} \mathbf{U}^{\mathcal{O}} \end{array} \right) \left(\begin{array}{c} \mathbf{U}^{\mathcal{H}} \\ \alpha_1 \\ \vdots \\ \alpha_{\mathsf{c}} \end{array} \right) = \left(\begin{array}{c} \mathbf{T}^{\mathcal{H}} \\ \mathbf{T}^{\mathcal{O}} \end{array} \right), \tag{2.17}$$

or, by defining vector $\boldsymbol{\alpha}$ and matrices $\boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}}$ and $\boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}}$ as

$$\left\{ \begin{array}{l} \boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_{\mathsf{c}})^T \\ \boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}} = (\mathbf{K}_1^{\mathcal{H}\mathcal{O}} \mathbf{U}^{\mathcal{O}}, \cdots, \mathbf{K}_{\mathsf{c}}^{\mathcal{H}\mathcal{O}} \mathbf{U}^{\mathcal{O}}) \\ \boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}} = (\mathbf{K}_1^{\mathcal{O}\mathcal{O}} \mathbf{U}^{\mathcal{O}}, \cdots, \mathbf{K}_{\mathsf{c}}^{\mathcal{O}\mathcal{O}} \mathbf{U}^{\mathcal{O}}) \end{array} \right. , \tag{2.18}$$

the previous system can be rewritten as

$$\left(\begin{array}{cc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}} \end{array} \right) \left(\begin{array}{c} \mathbf{U}^{\mathcal{H}} \\ \boldsymbol{\alpha} \end{array} \right) = \left(\begin{array}{c} \mathbf{T}^{\mathcal{H}} \\ \mathbf{T}^{\mathcal{O}} \end{array} \right), \tag{2.19}$$

that represents an overdetermined nonlinear algebraic system.

By premultiplying by the transpose of the matrix, a square algebraic system is obtained,

$$\left(\begin{array}{cc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}} \end{array} \right)^T \left(\begin{array}{cc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}} \end{array} \right) \left(\begin{array}{c} \mathbf{U}^{\mathcal{H}} \\ \boldsymbol{\alpha} \end{array} \right) = \left(\begin{array}{cc} \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{H}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{H}\mathcal{O}} \\ \sum\limits_{i=1}^{\mathsf{c}} \alpha_i \mathbf{K}_i^{\mathcal{O}\mathcal{H}} & \boldsymbol{\kappa}^{\mathcal{O}\mathcal{O}} \end{array} \right)^T \left(\begin{array}{c} \mathbf{T}^{\mathcal{H}} \\ \mathbf{T}^{\mathcal{O}} \end{array} \right),$$
$$\tag{2.20}$$

that allows us to calculate $\mathbf{U}^{\mathcal{H}}$ and $\boldsymbol{\alpha}$ by using an adequate nonlinear solver (e.g. fixed point, Newton, etc.).

When considering a linear behavior the resulting displacements, strains and stresses can easily be derived from $\mathbf{U} = (\mathbf{U}^{\mathcal{H}}, \mathbf{U}^{\mathcal{O}})^T$ by considering

$$
\begin{cases}
\mathbf{u}(\mathbf{x}) = \sum\limits_{i=1}^{\text{nd}} \mathbf{U}_i N_i(\mathbf{x}) \\
\boldsymbol{\varepsilon} = \nabla_s \mathbf{u} \\
\mathbf{C} = \sum\limits_{i=1}^{\text{c}} \alpha_i \mathbf{M}_i \\
\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}
\end{cases} \quad , \tag{2.21}
$$

where nd is the number of nodes considered to approximate the displacement field $\mathbf{u}(\mathbf{x})$, $N_i(\mathbf{x})$ the associated shape functions and $\nabla_s(\bullet)$ denotes the symmetric component of the gradient

2.2.1.2 Nonlinear elastic behavior

In the nonlinear case a major difficulty appears: since the constitutive behavior depends on strain, and it can be different at each physical point $\mathbf{x} \in \Omega$, the procedure just proposed and described to address the linear case must be adapted accordingly. As proposed in [Crespo *et al.* 2017] a simple and still appealing possibility to describe the constitutive manifold consists of approximating it in an adequate polynomial basis. The simplest alternative consists of approximating the elastic energy (as a function of the strain) whose first derivative results in the stress tensor and the second one leads to the elastic tensor.

Even though proceeding to identify the energy density functional seems to be a better alternative than identifying the elastic tensor, since it ensures thermomechanical consistency, and thus all symmetries associated to the material behavior. It is hard to find a basis which is always positive definite and concave for any possible evaluation of its arguments. Instead, we will parameterize each one of the α_i parameters appearing in Eq. 2.13 as a function of the strain state i.e. $\alpha_i \to \alpha_i(\boldsymbol{\varepsilon})$. It is important to highlight that the required symmetries of the elastic tensor $\mathbf{C}(\boldsymbol{\varepsilon})$ are strongly imposed with an appropiate choice of matrices $\mathbf{M}_i$ i.e. symmetric matrices.

The choice of the approximation basis deserves some comments. Imagine for a while the approximation of a one-dimensional function $f(\xi)$ in $\mathcal{I} = [\xi^-, \xi^+]$. A natural possibility consists of using piecewise continuous linear functions $N_i(\xi)$ to define its approximation, as it is usual within the finite element framework, by considering a mesh composed of q nodes uniformly distributed in $\mathcal{I}$, with coordinates ξ_i, $i = 1, \ldots, $q ($\xi_1 = \xi^-$ and $\xi_{\text{q}} = \xi^+$), from which the approximation reads

$$
f(\xi) = \sum_{i=1}^{\text{q}} f(\xi_i) N_i(\xi), \tag{2.22}
$$

where $N_i(\xi)$, for $1 < i < $q writes

$$
N_i(\xi) = \begin{cases}
\frac{\xi - \xi_{i-1}}{\xi_i - \xi_{i-1}} & \text{if } \xi \in [\xi_{i-1}, \xi_i] \\[2mm]
\frac{\xi_{i+1} - \xi}{\xi_{i+1} - \xi_i} & \text{if } \xi \in [\xi_i, \xi_{i+1}] \\[2mm]
0 & \text{elsewhere}
\end{cases} \quad , \tag{2.23}
$$

$$
N_1(\xi) = \begin{cases}
\frac{\xi_2 - \xi}{\xi_2 - \xi_1} & \text{if } \xi \in [\xi_1, \xi_2] \\[2mm]
0 & \text{elsewhere}
\end{cases} \quad , \tag{2.24}
$$

and

$$N_{\mathsf{q}}(\xi) = \begin{cases} \frac{\xi - \xi_{\mathsf{q}-1}}{\xi_{\mathsf{q}} - \xi_{\mathsf{q}-1}} & \text{if } \xi \in [\xi_{\mathsf{q}-1}, \xi_{\mathsf{q}}] \\ \\ 0 & \text{elsewhere} \end{cases}. \tag{2.25}$$

If the solution is known at different positions Ξ_j, $j = 1, \ldots, \mathsf{j}$, Eq. (2.22) will read

$$f(\Xi_j) = \sum_{i=1}^{\mathsf{q}} f(\xi_i) N_i(\Xi_j), \quad j = 1, \ldots, \mathsf{j}, \tag{2.26}$$

that results in the linear system

$$\begin{pmatrix} N_1(\Xi_1) & \ldots & N_{\mathsf{q}}(\Xi_1) \\ \vdots & \ddots & \vdots \\ N_1(\Xi_{\mathsf{j}}) & \cdots & N_{\mathsf{q}}(\Xi_{\mathsf{j}}) \end{pmatrix} \begin{pmatrix} f(\xi_1) \\ \vdots \\ f(\xi_{\mathsf{q}}) \end{pmatrix} = \begin{pmatrix} f(\Xi_1) \\ \vdots \\ f(\Xi_{\mathsf{j}}) \end{pmatrix}. \tag{2.27}$$

At this point, different situations can be found:

- An undetermined system if $\mathsf{j} < \mathsf{q}$;

- A determined one, if $\mathsf{j} = \mathsf{q}$;

- An overdetermined one, if $\mathsf{j} > \mathsf{q}$. However, even when $\mathsf{j} \geq \mathsf{q}$ the resulting system can become undetermined if at least for one node ξ_i, no point Ξ_j, $\forall j$, falls in its support, $[\xi_{i-1}, \xi_{i+1}]$.

In these circumstances different algebraic solutions exist (e.g. pseudo-inverse, "matlab backslash", L^2 or L^1 optimization, ...). However, in this work we decided to consider global approximation functions in $[\xi^-, \xi^+]$. To avoid the issues related to high-order Lagrange approximations, we consider approximations based on the use of orthogonal polynomials, and more precisely Chebyshev polynomials. Thus, Eq. (2.22) is replaced by

$$f(\xi) = \sum_{i=1}^{\mathsf{q}} \gamma_i T_i(\xi), \tag{2.28}$$

where $T_i(\xi)$ refer to Chebyshev polynomials and the weights γ_i are computed from its associated linear system

$$f(\Xi_j) = \sum_{i=1}^{\mathsf{q}} \gamma_i T_i(\Xi_j), \quad j = 1, \ldots, \mathsf{j}, \tag{2.29}$$

where singularity issues are circumvented as soon as $\mathsf{j} \geq \mathsf{q}$ and there are not repeated points.

However, problems arise as soon as the approximation becomes multidimensional. This is the case when approximating a scalar function f (or equivalently each α_i) as a function of the 6 components of the strain tensor $\boldsymbol{\varepsilon}$, using the same degree (q) for each component. In this case, the approximation

$$f(\boldsymbol{\varepsilon}) \approx \sum_{ijklmn}^{\mathsf{q}^6} \gamma_{ijklmn} T_i(\varepsilon_{11}) T_j(\varepsilon_{12}) T_k(\varepsilon_{13}) T_l(\varepsilon_{22}) T_m(\varepsilon_{23}) T_n(\varepsilon_{33}), \tag{2.30}$$

contains too many coefficients γ_{ijklmn} (in fact q^6), and consequently the accuracy requires the same number of data points (even if sparse sampling could be an appealing

alternative). Of course the approximation could be limited to a certain degree $\mathcal{D}$ by considering in the previous sum indexes verifying $i + j + k + l + m + n \leq \mathcal{D}$.

An alternative approximation makes use of a separated representation (usually considered within the proper generalized decomposition (PGD) framework [Chinesta *et al.* 2010, Chinesta *et al.* 2011b]) that reads

$$f(\varepsilon) \approx \sum_m^{\texttt{M}} E_m^{11}(\varepsilon_{11}) \, E_m^{12}(\varepsilon_{12}) \, E_m^{13}(\varepsilon_{13}) \, E_m^{22}(\varepsilon_{22}) \, E_m^{23}(\varepsilon_{23}) \, E_m^{33}(\varepsilon_{33}). \tag{2.31}$$

where M is the number of modes required to express the scalar function in a separated format. This separated representation is specially appropriate when ε is defined in the hyper-hexahedral domain $\mathcal{E} = [\varepsilon_{11}^-, \varepsilon_{11}^+] \times [\varepsilon_{12}^-, \varepsilon_{12}^+] \times \cdots \times [\varepsilon_{33}^-, \varepsilon_{33}^+]$. However, admissible deformations imply non separable domains. The application of separated representation in non-separable domains was deeply addressed in [Gonzalez *et al.* 2010] where the use of R-functions succeeded to represent complex non-separable geometries.

To avoid singularity issues, functions E_m^{kl}, are approximated by using global Chebyshev polynomials, according to

$$E_m^{kl}(\varepsilon_{kl}) \approx \sum_{j=1}^{\texttt{q}_{kl}} \gamma_j^{kl,m} T_j(\varepsilon_{kl}) = \boldsymbol{\gamma}^{kl,m} \cdot \mathbf{T}^{kl} \tag{2.32}$$

Starting from the weak form

$$\int_\Omega \boldsymbol{\varepsilon}^* \cdot \boldsymbol{\sigma} \, d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{t} \, d\mathbf{x}, \tag{2.33}$$

we substitute the constitutive relationship

$$\int_\Omega \boldsymbol{\varepsilon}^* \cdot \mathbf{C}(\varepsilon)\boldsymbol{\varepsilon} \, d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{t} \, d\mathbf{x}, \tag{2.34}$$

and the non-linear parameterization of the constitutive elastic tensor

$$\sum_i^{\texttt{c}} \int_\Omega \alpha_i(\varepsilon)\boldsymbol{\varepsilon}^* \cdot \mathbf{M}_i \boldsymbol{\varepsilon} \, d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{t} \, d\mathbf{x}, \tag{2.35}$$

Where each $\alpha_i(\varepsilon)$ coefficient is given by

$$\alpha_i(\varepsilon) \approx \alpha_i^M(\varepsilon) = \sum_m^{\texttt{M}} E_{i,m}^{11}(\varepsilon_{11}) \, E_{i,m}^{12}(\varepsilon_{12}) \, E_{i,m}^{13}(\varepsilon_{13}) \, E_{i,m}^{22}(\varepsilon_{22}) \, E_{i,m}^{23}(\varepsilon_{23}) \, E_{i,m}^{33}(\varepsilon_{33}). \tag{2.36}$$

Note that functions E_{im}^{kl}, as well as the unknown nodal displacements, should be computed from the knowledge of the measurable nodal displacements accessible in a part of the domain Ω, as was the case in the procedures discussed previously. Moreover, since the constitutive elastic tensor depends on the strain state, an initial guess for the hidden displacement field $(\mathbf{U}_0^{\mathcal{H}})$ must be given to compute a consistent initial guess of the initial strain state ε_0. Hence, the Chebychev functions evaluated at ε_0, namely $\mathbf{T}_0^{kl}$, are computable.

As in the case of the PGD constructor, we consider a greedy algorithm that computes sequentially these functions [Chinesta *et al.* 2010]. Thus at PGD iteration $M+1$, we assume that the rank-M approximation of the elastic tensor $\mathbf{C}^M$ was already computed, i.e.,

$$\mathbf{C}(\varepsilon) \approx \mathbf{C}^M(\varepsilon) = \sum_i^c \alpha_i^M(\varepsilon)\mathbf{M}_i. \tag{2.37}$$

At present iteration we look for the new functional product leading to the updated enriched rank-$M+1$ expression of $\mathbf{C}^{M+1}(\varepsilon)$ from

$$\mathbf{C}^{M+1}(\varepsilon) = \mathbf{C}^M(\varepsilon) + \sum_i^c E_{i,M+1}^{11}(\varepsilon_{11})\ldots E_{i,M+1}^{33}(\varepsilon_{33})\mathbf{M}_i = \mathbf{C}^M(\varepsilon) + \Delta\mathbf{C}(\varepsilon), \tag{2.38}$$

that introduced into the weak form results

$$\int_\Omega \varepsilon^* \cdot \left(\mathbf{C}^M(\varepsilon) + \Delta\mathbf{C}(\varepsilon)\right)\varepsilon \; d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{t} \; d\mathbf{x}. \tag{2.39}$$

As is the case when applying the PGD solver, the solution procedure consists of using an alternated direction fixed point strategy, that proceeds as follows [Chinesta *et al.* 2010]:

1. By considering $E_{i,M+1}^{12(r-1)}, \ldots, E_{i,M+1}^{33(r-1)}$ from the previous fixed point iteration $r-1$ of the nonlinear solver (initialized at $r=1$ from the functions at the previous enrichment iteration M), we compute $E_{i,M+1}^{11(r)}$.

2. The process is repeated but now with $E_{i,M+1}^{11(r)}, E_{i,M+1}^{13(r-1)}, \ldots, E_{i,M+1}^{33(r-1)}$ known. This allows to compute the unknown nodal displacements and functions involved in $E_{i,M+1}^{12(r)}$. The process is repeated for all the other components until computing $E_{i,M+1}^{33(r)}$. Then, the fixed point convergence criterion is checked and if it is not attained we move to the next fixed point iteration $r+1$.

3. When reaching the fixed point convergence, the enrichment convergence is evaluated and if it is not attained we move to the next elastic tensor approximate $\mathbf{C}^{M+2}$ from the just competed $\mathbf{C}^{M+1}$. It is important to highlight that when the enrichment of the elastic tensor has finished, a new guess of the hidden displacement field is available($\mathbf{U}_1^{\mathcal{H}}$), and the Chebychev functions are evaluated again $\mathbf{T}_1^{kl}$. We assume that at iteration $\mathbf{p}$ the global non-linear enrichment process converges and consequently we have access to both the hidden displacement field and the elastic tensor manifold from which simulations can be carried out as described in section 2.1.

Remark. It is important to note that data coming from $\mathcal{S}$ different snapshoots can be joined, as soon as $\mathcal{S}$ weak balance laws are formulated. The common link between these different balance laws is that they share the same constitutive relationship i.e. same $\alpha_i(\varepsilon)$ for each of the snapshoots.

2.2.2   Numerical results

To illustrate the capabilities of the just described procedure, we consider the simple mechanical problem depicted in Fig. 2.12. It consists of a two-dimensional unit squared solid, $\mathbf{x} = (x,y) \in \Omega = (0,1) \times (0,1)$, equipped with a nonlinear elastic material, clamped along its basis $y = 0$, free of traction on its lateral boundaries $x = 0$ and $x = 1$ and with a uniformly distributed traction $\mathbf{t}$ on its upper boundary $y = 1$. We analyze the performance of the inverse identification procedure.
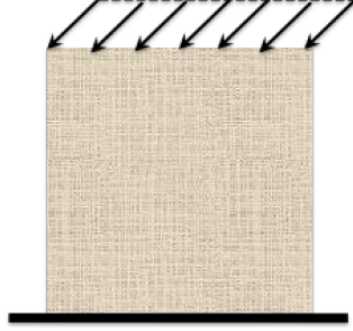
Figure 2.12 – Schema of the considered mechanical problem

2.2.2.1 Synthetic generation of displacement measures

In the present case, we consider an applied traction whose orientation, i.e. $\mathbf{t} = t\mathbf{p}$, is arbitrary: $\mathbf{p}(\theta) = (\cos\theta, \sin\theta)^T$, $\theta \in [0, 2\pi)$. As just discussed, different traction magnitudes and orientations are tested to have a reasonable data-base of snapshoots ($\mathcal{S}$) to explore the constitutive manifold. From

$$t = \sum_{j=1}^{\ell} \Delta t_j, \tag{2.40}$$

we can define an intermediate traction magnitude at step r, t_r, from

$$t_r = \sum_{j=1}^{r} \Delta t_j, \tag{2.41}$$

that leads to different tractions depending on the orientation

$$\mathbf{t}_r^s = \sum_{j=1}^{r} \Delta t_j \mathbf{p}(\theta^s), \tag{2.42}$$

$$\theta^s = \sum_{m=1}^{s} \Delta\theta, \tag{2.43}$$

with R defining the angular discretization, that is the number of discrete angles considered,

$$R\Delta\theta = 2\pi. \tag{2.44}$$

To discretize the mechanical problem, the domain Ω was equipped with a uniform mesh consisting of $P \times P$ square finite elements, where a bilinear approximation of the displacement field was considered. The displacement is therefore assumed to be experimentally measurable at some nodes, namely $\mathbf{U}^{\mathcal{O}}$. It is important to reckon that the more observable points are introduced in the algorithm, the better is the inverse constitutive identification. Or similarly, increasing the number of snapshoots also increases the accuracy in the identification.

In order to generate pseudo-experimental displacement measurements, we consider a general nonlinear elastic behavior of the type (Voigt notation is employed here)

$$\mathbf{C} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{2.45}$$
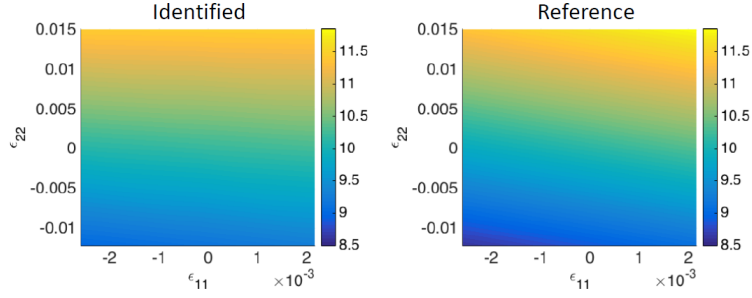
Figure 2.13 – Identified (left) versus real (right) $\mathbf{C}_{11}$ component in the trace space.

with the elastic coefficients given by

$$
\begin{cases}
E = E_0 + E_1 \mathrm{Tr}(\boldsymbol{\varepsilon}) \\
\nu = \nu_0 + \nu_1 \mathrm{Tr}(\boldsymbol{\varepsilon})
\end{cases} ,
\tag{2.46}
$$

with E_0, E_1, ν_0 and ν_1 positive constants and where $\mathrm{Tr}(\bullet)$ refers to the trace operator acting on tensor $\bullet$. Coefficients ν_0, ν_1 and E_0, E_1 where selected such that $\nu \in (0, 0.5)$ and $E > 0$ in the range of deformations considered.

Two synthetic constitutive manifolds have been tested, the linear case i.e. $E_0 = 10$, $\nu_0 = 0.1$ and $E_1 = \nu_1 = 0$. Whereas the non-linear case parameters were selected as $E_0 = 10$, $\nu_0 = 0.1$, $E_1 = 10$ and $\nu_1 = 0.1$. The applied tension was $t = 0.1$ and it was applied by considering 10 loading steps, i.e. $\ell = 10$ and 10 orientations, i.e. $R = 10$. The mesh consisted of 10×10 $Q1$ finite elements allowing the calculation of the displacement at each loading step.

2.2.2.2   Unveiling the constitutive manifold

The fact that the constitutive law employed to generate pseudo-experimental displacements was known is now forgotten, and the behavior is assumed unknown from now on. The main objective is therefore to determine the constitutive manifold of the material, that is, its sampling stress-strain couples, with the only information provided by the mechanical test illustrated in Fig. 2.12 and the recorded displacements at the 11 locations at each loading step.

For this purpose we proceed as described in section 2.2.1 to identify the non-linear relationship. When considering $\forall k, l$, $\mathsf{q}_{kl} = \mathsf{q} = 5$ and only one load applied on the top boundary $\mathbf{t} = 0.1\mathbf{p}$, with $\mathbf{p} = (-1/\sqrt{2}, -1/\sqrt{2})$, and by assuming that our pseudo-experimental technique is able to provide us with nodal displacement values in the 25% of the nodal locations in the model, a good agreement was obtained between the identified and the reference behavior as proved in Fig. 2.13. The identified (left) versus the real (right) of $\mathbf{C}_{11}$ in the ϵ_{11}, ϵ_{22} space is plotted. As it can be seen, there is a good agreement between the identified and real constitutive behaviours.

Fig. (2.14) provides a deeper understanding of the inverse identification procedure. The colormap indicates the relative error of the $\mathbf{C}_{11}$ component in the trace space. The black points correspond to the strain state for every Gauss points inside the material. As it can be noticed, bottom-left and top-right corners do not have black point in its vicinity. Therefore, it is normal to expect a worse identification is these areas. To alleviate this problem, it is possible to combine several snapshoots in such a way that there are also strain states in the vicinity of these areas.
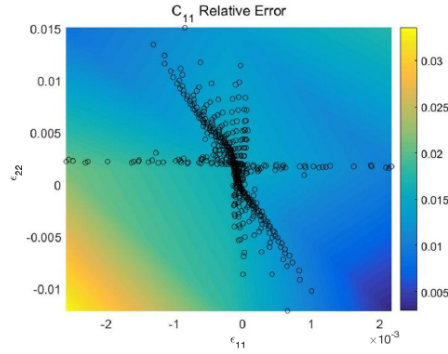
Figure 2.14 – Relative error related to the component $\mathbf{C}_{11}$. Black points indicate the strains that were available for the identification procedure.

The reduction of the number of measured displacements requires the use of additional loading test cases. We also proved that the convergence is significantly enhanced with the number of considered loading cases, the number of measured displacements and the considered polynomial degree for approximating the behavior. Moreover, the use of a separated representation allows to diminish the number of experimental measurements because it involves optimal polynomial representations of the constitutive manifold.

Figures 2.15, 2.16 and 2.17 compare the different identified components of the stress tensor, $\boldsymbol{\sigma}_{xx}$, $\boldsymbol{\sigma}_{yy}$ and $\boldsymbol{\sigma}_{xy}$ and the reference ones obtained from Eq. (2.45). The stress magnitude in those figures is represented from the color bar. These figures reveal an almost perfect stress-strain couple match with the pseudo-experimental ones, with relative errors lower than 1%.

It was proved, that as expected, by decreasing the loading step, that is, by increasing ℓ and R in the loadings expressed from Eqs. (2.40) and (2.44), the error with respect to the reference one (related to the constitutive equation (2.45)) decreases proving the expected convergence of the proposed inverse identification strategy as Fig. 2.18 reveals. Errors fewer than few percent using the norm $\|\mathbf{C}^{identified} - \mathbf{C}\|_2$, are easily reachable.

To further explore the characteristics of the identified constitutive manifold, we decided to apply a non-linear dimensionality reduction technique to the stress-strain couples just obtained. By applying on them Locally Linear Embedding nonlinear dimensionality reduction strategy, see Figure 2.19, we compare the dimensionality of the resulting linear and nonlinear constitutive manifolds. In the linear case two parameters seemed to be enough for visualizing and parametrizing the constitutive data (this number corresponds with the number of lowest eigenvalues before reaching the typical plateau of LLE techniques [Roweis & Saul 2000], see Fig. 2.19). However, when considering the manifold that results from the identified stress-strain couples describing the nonlinear case, the dimensionality seems to increase to three parameters. This is natural since the nonlinear behavior implies the need for more complex descriptions.

Alternatively, we employed k-PCA nonlinear dimensionality reduction [González *et al.* 2016], that allows to visualize low-dimensional manifolds within a higher dimensional space. When applying k-PCA to the identified data corresponding to the nonlinear behavior, we obtain the embedding depicted in Fig. 2.20. Here, in order to prove that the embedded, low-dimensional data is well distributed on the slow manifold, we assigned a color to each data point corresponding to its elastic energy. In order to prove that the reduced data defined an almost perfect 2D manifold, we represent in different
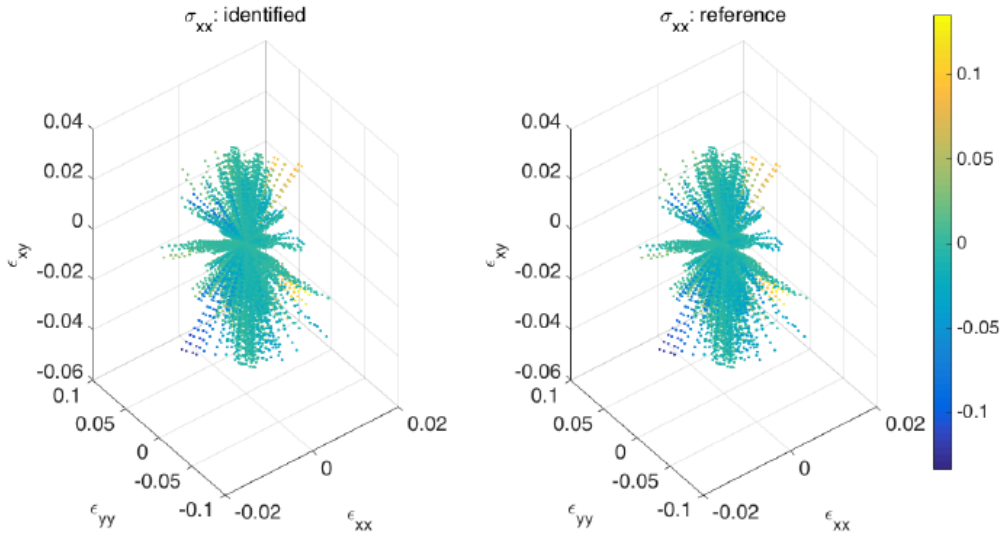
Figure 2.15 – $\boldsymbol{\sigma}_{xx} = \boldsymbol{\sigma}_{xx}(\boldsymbol{\varepsilon})$: identified from data (left) and reference values (right).

views in Fig 2.20 of the same reduced manifold.

Data-driven simulation   The nonlinear elastic problem is now solved by employing the constitutive manifold just identified, when a traction $\mathbf{t} = t\mathbf{p}$, $\mathbf{p}^T = (\cos 3\pi/2, \sin 3\pi/2)$ applies on the top boundary. The reference displacement field calculated with the constitutive model (2.45)-(2.46) is depicted in Fig. 2.21 and compared with the one obtained when solving the same problem but now with the identified constitutive model whose solution is depicted in Fig. 2.22. Both results are in good agreement despite the coarse descriptions considered.

2.2.3   Conclusions

We proved in section 2.1 that numerical simulations can be performed from the only knowledge of data defining the material behavior. It was claimed that the main drawback of one such approach is the necessity of unveiling the whole constitutive manifold. However, at present, testing facilities are not able to explore the whole strain-stress space in a continuous way. In this work we considered elastic behaviors (linear and nonlinear), proving that the constitutive manifold can be extracted from a data-driven inverse procedure in an effective manner.

A procedure involving the polynomial approximation of the whole constitutive manifold, seems to be robust from all points of view, and its immersion in a hierarchical or multi-resolution strategy seems an appealing choice for future developments. Even if the results only concerned some simplistic behaviors, the methodology seems to be appropriate to address more complex scenarios, such as behaviors involving large strains, as well as inelastic deformations. Other points that should be considered are the ones related to existence and propagation of noise. This constitutes our current effort of research.
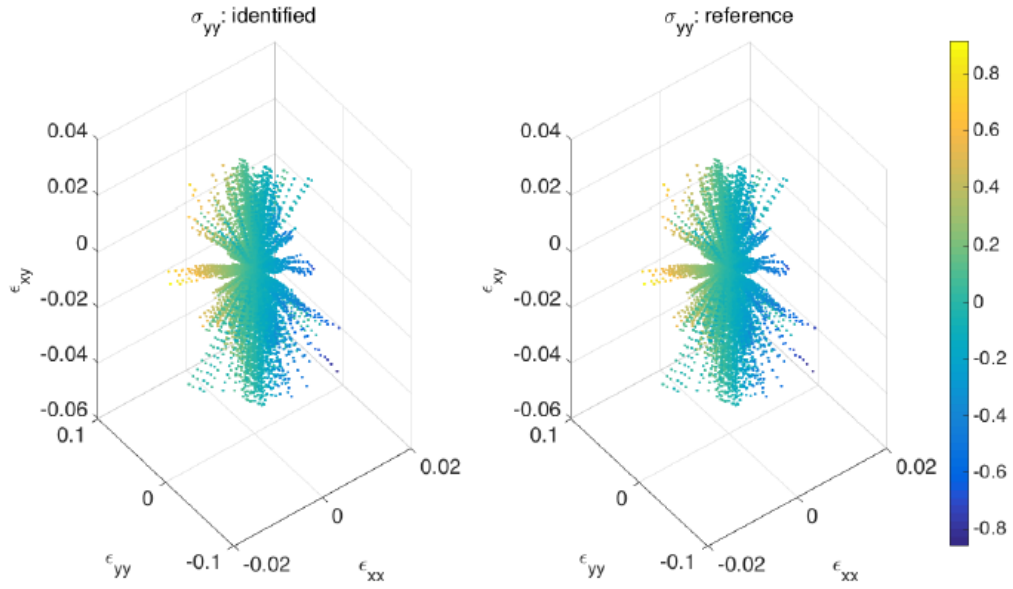
Figure 2.16 – $\boldsymbol{\sigma}_{yy} = \boldsymbol{\sigma}_{yy}(\boldsymbol{\varepsilon})$: identified from data (left) and reference values (right).
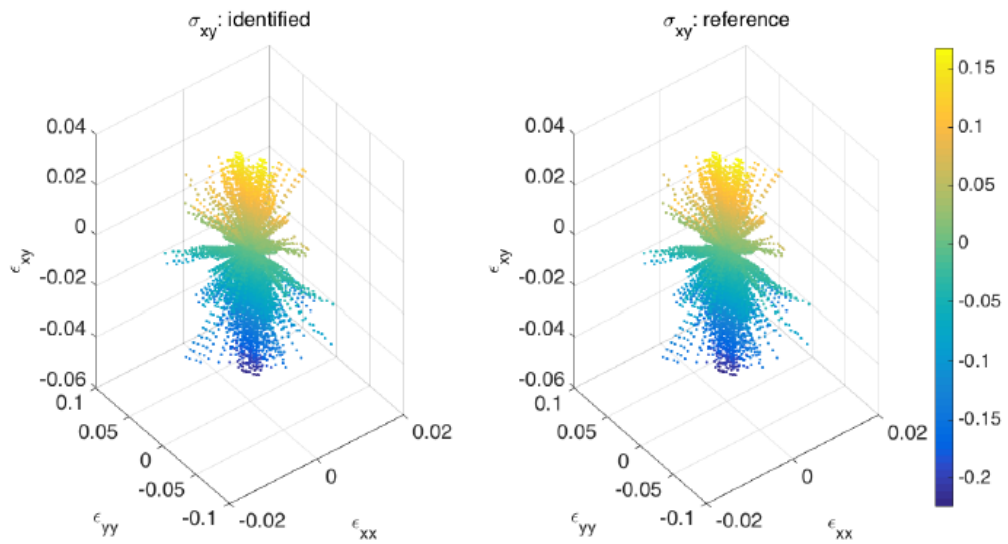
Figure 2.17 – $\boldsymbol{\sigma}_{xy} = \boldsymbol{\sigma}_{xy}(\boldsymbol{\varepsilon})$: identified from data (left) and reference values (right).
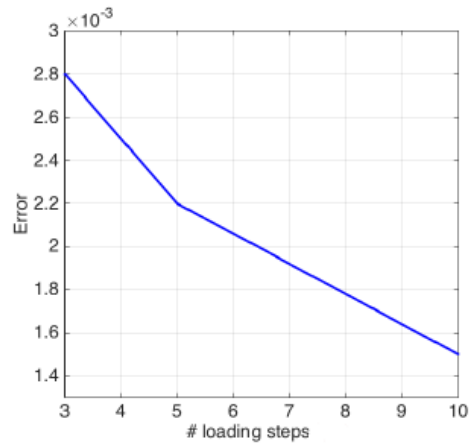
Figure 2.18 – Evolution of the error with the loading steps ℓ
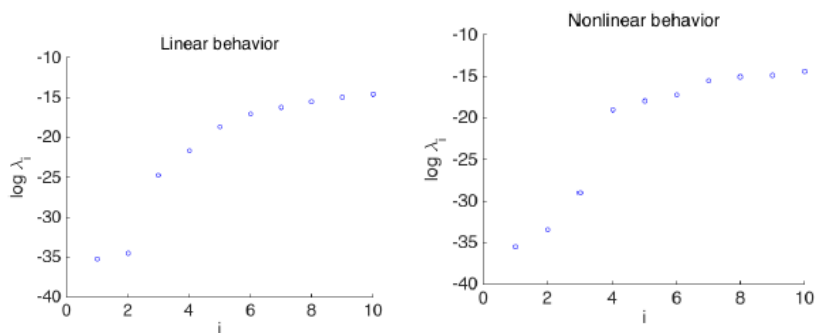
Figure 2.19 – Dimensionality of the linear (left) and nonlinear (right) manifolds when applying the LLE nonlinear dimensionality reduction technique on the stress-strain data.
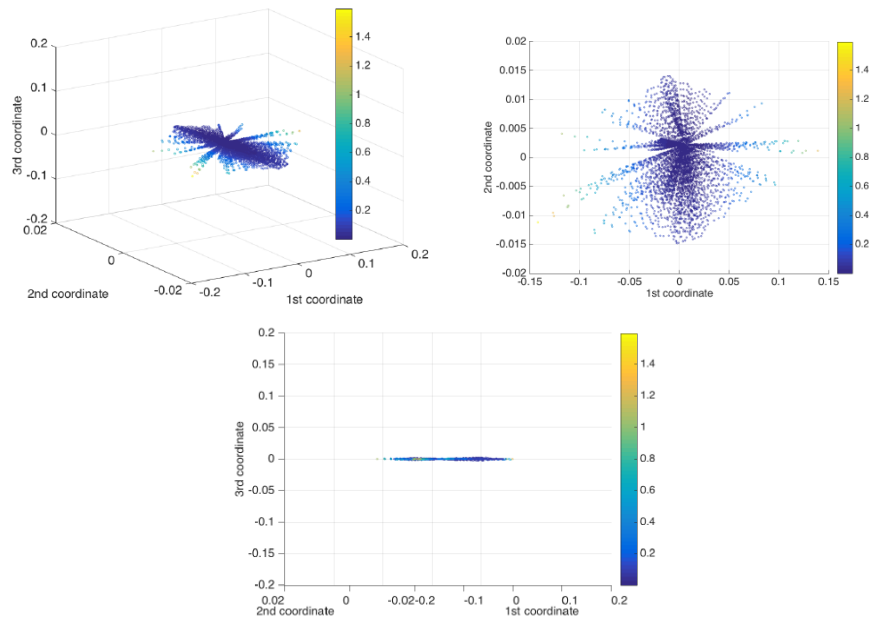
Figure 2.20 – Different views of the constitutive manifold represented in a reduced 3D space.
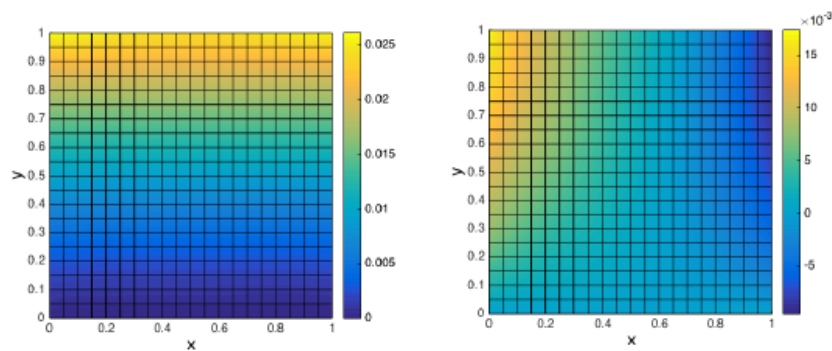
Figure 2.21 – Data-driven simulation based on the reference constitutive manifold. Displacement field: $\mathbf{u}_x$ (left) and $\mathbf{u}_y$ (right)
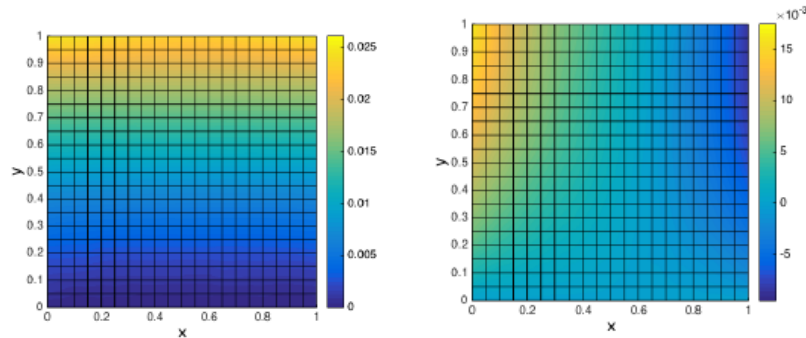
Figure 2.22 – Data-driven simulation based on the identified constitutive manifold. Displacement field: $\mathbf{u}_x$ (left) and $\mathbf{u}_y$ (right).

2.3   From elastic homogeneization to upscaling of non-Newtonian fluid flows in porous media

In section 2.1, we showed that it is possible to perform numerical simulations directly from data by circumventing, or at least alleviating, the necessity of considering an explicit constitutive model to relate kinematic and dynamics variables. Thus, simulations proceed directly from the stress-strain data, from which standard discretization techniques proceed. In this section, we perform a step forward in order to define macroscopic constitutive manifolds from the microscopic calculations, and then solve the macroscopic mechanical problem from the data contained in these constitutive manifolds without having to determine explicitly an homogenized or upscaled constitutive equation. Particularly, the upscaling of non-Newtonian fluids in porous media will be analyzed.

When considering elastic models, where the different phases involved in the microstructure exhibit the same mechanical behaviour, whether linear or non-linear, homogenization is straightforward. Two main approaches exists, the hierarchical and the concurrent discussed in [Borst 2008, Fish 006, McVeigh & Liu 2008, Lamari *et al.* 2010]. In the hierarchical approach micro and macro calculations are decoupled [qnd D. Dureisseix & Cartraud 2009, Fish & Yuan. 2005, Strouboulis *et al.* 2004] whereas in the concurrent one both are strongly coupled [Kanoute *et al.* 2009, Feyel 1999, Geers *et al.* 2010, J.C. Michel 2004, I. Temizer 2007a, I. Temizer 2007b, Yvonnet *et al.* 2009, Halabi *et al.* 2013].

The standard macro-scale theoretical framework for flows in porous media is the Darcy model which relates averaged fluid velocities to pressure gradients. Being purely dissipative, the Darcy model cannot address elastic effects present in viscoelastic fluids. The same issue arises in modelling the flow of suspensions. While suitable evolution equations can be formulated for the micro-scale conformation, this cannot be achieved for averaged conformational quantities at the macro-scale. In such circumstances, a possible route consists in defining a macroscopic flow model able to incorporate elastic effects for addressing viscoelastic fluids or an evolution equation for the upscaled conformation in the case of suspensions. Such a route, however, has two main handicaps: its intrinsic difficulty and its incompatibility with essentially all available simulation software used in industrial applications, which make use of Darcy's model.

Thus, the most valuable route consists in considering as much as possible viscous models that are more or less complex to capture the main fluid and flow features at the microscopic scale, and from them obtaining (by upscaling) an effective Darcy permeability at the macro-

scopic scale to be given as input to conventional simulation software. The main issue then is the calculation of the effective permeability. To do so, in this work as in many others, we enforce equality of the dissipated powers at both the micro and macro-scales.

In former works such as [Ammar *et al.* 2016, Lopez *et al.* 2015, Lopez *et al.* 2016c], the upscaling route was successfully applied when considering purely viscous behaviors (Newtonian and generalized-Newtonian fluids) or by addressing viscoelastic behaviors from quasi-Newtonain purely viscous formulations. In general, for non-linear behaviours, however, this approach quickly becomes very intricate and computationally expensive.

In this section, we follow a similar rationale, but instead of looking for an explicit expression of the effective permeability, we propose to build the so-called dissipation manifold, whose second derivative with respect to the macroscopic velocity directly yields the manifold of effective permeabilities.

This section is organized as follows: subsection 2.3.1 introduces the main features behind homogenization, upscaling and its associated macroscopic simulations. Subsection 2.3.2 introduces the constitutive modelling of non-linear viscous fluids in porous media together with the manifold-based homogenization procedure. Subsection 2.3.3.1 provides numerical results associated to two different kind of microstructures.

2.3.1 Revisiting homogenization, upscaling and macroscopic simulation

First, we consider the mechanical problem defined in a domain Ω occupied by a heterogeneous elastic material involving a number of phases evolving very fast spatially, thus allowing for scale separation. Even if the behaviour of each phase is assumed perfectly defined, the solution of the elastic problem in the whole domain requires a very fine discretization mesh for approximating the different fields (displacement, strain and stress) able to capture the microscopic details. To circumvent this difficulty, a widely considered approach consists in calculating the homogenized behaviour of the material by proceeding at the microscopic scale within the so-called representative volume element − RVE − ω, and then using these homogenized properties in the macroscopic calculation with a coarse mesh size larger than the characteristic length of the microstructure.

Assuming a linear elastic behaviour for each phase coexisting in the composite material, the Cauchy stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$ are related at each point $\mathbf{x} \in \omega$ by the constitutive model $\boldsymbol{\sigma} = \mathbf{c} : \boldsymbol{\varepsilon}$, where $\mathbf{c}$ embodies the micro-scale elastic properties. Without loss of generality, the same microstructure is assumed everywhere in Ω. Thus, we can define the macroscopic strain $\mathbf{E}$ and stress $\boldsymbol{\Sigma}$ in ω according to the following spatial averages over the RVE:

$$\begin{cases} \mathbf{E} = \langle \boldsymbol{\varepsilon} \rangle = \frac{1}{|\omega|} \int_\omega \boldsymbol{\varepsilon}(\mathbf{x}) \, d\mathbf{x} \\ \boldsymbol{\Sigma} = \langle \boldsymbol{\sigma} \rangle = \frac{1}{|\omega|} \int_\omega \boldsymbol{\sigma}(\mathbf{x}) \, d\mathbf{x} \end{cases} . \tag{2.47}$$

Then, assuming the existence of a localization tensor $\mathbf{L}$ such that $\boldsymbol{\varepsilon}(\mathbf{x}) = \mathbf{L}(\mathbf{x}) : \mathbf{E}$, we obtain

$$\boldsymbol{\Sigma} = \langle \boldsymbol{\sigma} \rangle = \langle \mathbf{c}(\mathbf{x}) : \boldsymbol{\varepsilon}(\mathbf{x}) \rangle = \langle \mathbf{c}(\mathbf{x}) : \mathbf{L}(\mathbf{x}) \rangle : \mathbf{E}, \tag{2.48}$$

from which the homogenized elastic behavior $\mathbf{C}$ can be identified:

$$\mathbf{C} = \langle \mathbf{c}(\mathbf{x}) : \mathbf{L}(\mathbf{x}) \rangle. \tag{2.49}$$

The localization tensor can be computed in the linear case by solving three (in the 2D case) or six (in the 3D case) boundary value problems over ω, with affine displacements (that satisfy the Hill-Mandel principle) specified at the boundary $\partial \omega$ [Chinesta *et al.* 2008].

Obviously, the above rationale remains valid when the domain Ω contains different representative volume elements. It suffices to apply the procedure to each RVE, or in the limit case, at each evaluation point $\mathbf{X} \in \Omega$ (integration point of the macroscopic discretized problem) by considering its associated microstructure $\omega(\mathbf{X})$.

In the non-linear case, the computational implementation is slightly more complex as the behaviour depends on the macroscopic strain, and consequently the elastic homogenized tensor is a function of the considered point $\mathbf{X} \in \Omega$ as soon as the macroscopic strain varies in Ω even if the microstructure remains the same everywhere in Ω.

From a methodological view point, we can assume the existence of a homogenized elastic tensor at each location $\mathbf{X} \in \Omega$. We could solve the non-linear elastic problem in the RVE attached to each Gauss point considered for discretizing the homogenized elastic problem at the macroscopic scale, by enforcing the displacement in agreement with the existing macroscopic strain at that location. Then, the elastic properties at each position $\mathbf{x} \in \omega(\mathbf{X})$ could be frozen in order to linearize the problem in ω before applying the rationale described above for homogenizing the linear behaviour.

The main drawback of this approach is the computational cost. It can be significantly alleviated by means of an alternative strategy based on the Proper Generalized decomposition (PGD) [Chinesta *et al.* 2011a, Chinesta *et al.* 2013a, Chinesta *et al.* 2014]. In [Lamari *et al.* 2010], we proposed the calculation, for a given microstructure, of the parametric solution of the non-linear elastic problem in the REV for all feasible affine displacement enforced at its boundary.

A similar strategy was also successfully used in [Lopez *et al.* 2016c] when addressing the non-linear Stokes flow problem for a generalized-Newtonian fluid flowing in a porous medium. The resulting parametric local strain rates were used for determining the local parametric behaviour, and from it for obtaining the parametric solution of the boundary value problems, thus yielding a parametric localization tensor and from it the parametric upscaled Darcy behaviour.

Even though successful in these relatively simple problems, the above PGD procedure becomes very intricate to apply in more complex cases. In what follows, we develop an alternative approach that does not require the explicit evaluation of the homogenized or upscaled behaviours.

2.3.1.1 PGD-based generator of a macroscopic constitutive manifold

We consider a non-linear elastic problem defined in ω, and assume without loss of generality that the microstructure in ω represents the one existing everywhere in Ω. Within the PGD framework [Chinesta *et al.* 2011a, Chinesta *et al.* 2013a, Chinesta *et al.* 2014], we view the boundary conditions specified at $\partial\omega$ as model parameters and thus extra-coordinates of the problem.

Within the usual first-gradient elasticity framework, prescription of linear displacements at the boundary $\partial\omega$

$$\mathbf{u}(\mathbf{x} \in \partial\omega) = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} E_{11}x_1 + E_{12}x_2 + E_{13}x_3 \\ E_{12}x_1 + E_{22}x_2 + E_{23}x_3 \\ E_{13}x_1 + E_{23}x_2 + E_{33}x_3 \end{pmatrix}, \tag{2.50}$$

ensures the recovery of any macroscopic strain

$$\mathbf{E} = \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{12} & E_{22} & E_{23} \\ E_{13} & E_{23} & E_{33} \end{pmatrix}. \tag{2.51}$$

Within the PGD framework [Chinesta *et al.* 2011a, Chinesta *et al.* 2013a, Chinesta *et al.* 2014, Lamari *et al.* 2010], the coefficients E_{ij} in (2.50) are viewed as extra-coordinates, and we seek the parametric solution

$$\mathbf{u}(\mathbf{x}, E_{11}, E_{12}, \cdots, E_{33}) \approx$$

$$\sum_{i=1}^{N} \mathbb{X}_i(\mathbf{x}) \circ \mathbb{E}_i^{11}(E_{11}) \circ \mathbb{E}_i^{12}(E_{12}) \circ \mathbb{E}_i^{13}(E_{13}) \circ \mathbb{E}_i^{22}(E_{22}) \circ \mathbb{E}_i^{23}(E_{23}) \circ \mathbb{E}_i^{33}(E_{33}), \qquad (2.52)$$

where $\circ$ refers to the Hadamard product.

From the parametric displacement field (2.52), it is straightforward to compute the parametric strain and stress everywhere within ω, namely

$$\begin{cases} \boldsymbol{\varepsilon}(\mathbf{x}, E_{11}, E_{12}, E_{13}, E_{22}, E_{23}, E_{33}) \\ \boldsymbol{\sigma}(\mathbf{x}, E_{11}, E_{12}, E_{13}, E_{22}, E_{23}, E_{33}) \end{cases}, \qquad (2.53)$$

from which macroscopic strain as stress, $\mathbf{E}$ and $\boldsymbol{\Sigma}$ respectively, can be obtained from Eq. (2.47). As just indicated $\mathbf{E}$ is directly given by Eq. (2.51).

The macroscopic constitutive manifold $\boldsymbol{\Sigma}(\mathbf{E})$ is then defined by

$$\boldsymbol{\Sigma}(\mathbf{E}) = \langle \boldsymbol{\sigma}(\mathbf{x}, E_{11}, E_{12}, E_{13}, E_{22}, E_{23}, E_{33}) \rangle. \qquad (2.54)$$

Now, as many macroscopic strain-stress couples $(\boldsymbol{\Sigma}_m, \mathbf{E}_m)$, $m = 1, \ldots, M$ can be generated in real time by simply particularizing the parametric solution (2.54). Each stress-strain couple is a single point $\mathbf{P}_m$ in a space of dimension $D = 12$ (the six distinct components of the stress and strain tensors, respectively). In the sequel, we use Voigt's notation, i.e. stress and strain tensors will be represented as vectors and consequently the fourth-order elastic tensor reads as a 6×6 square matrix.

Each vector $\mathbf{P}_m$ thus defines a point in a space of dimension D and, therefore, the whole set of stress-strain couples is a set of M points in $\mathbb{R}^D$. We conjecture that all these sample points belong to a low-dimensional manifold embedded in the high-dimensional space $\mathbb{R}^D$, thus allowing for a non-linear dimensionality reduction as discussed in section 2.1. In what follows, however, we proceed without such a dimensionality reduction and use the simplest strategy proposed and discussed in section 2.1. We consider locally-linear approximations, that allow us to write

$$\mathbf{P}_m = \sum_{i=1}^{M} W_{mi} \mathbf{P}_i, \qquad (2.55)$$

with $W_{mi} = 0$ if $i \notin \mathcal{S}_m$, being $\mathcal{S}_m$ the set containing the K-nearest neighbours of $\mathbf{P}_m$. By minimizing the functional

$$\mathcal{H}(\mathbf{C}) = \sum_{i \in \mathcal{S}_m} (\boldsymbol{\Sigma}_i - \mathbf{C} \cdot \mathbf{E}_i)^2, \qquad (2.56)$$

we obtain the locally-linear behaviour $\mathbf{C}(\mathbf{P}_m) \equiv \mathbf{C}_m$.

2.3.1.2 Manifold-based simulation

Once the locally-linear behaviour $\mathbf{C}(\mathbf{P})$ is identified, we may apply the simplest linearization technique operating on the standard weak form

$$\int_\Omega \mathbf{E}^*(\mathbf{X}) : \boldsymbol{\Sigma}(\mathbf{X}) \, d\mathbf{X} = \int_{\Gamma_N} \mathbf{U}^*(\mathbf{X}) \cdot \mathbf{T}(\mathbf{X}) \, d\mathbf{X}, \qquad (2.57)$$

where at each point, from the stress-strain couple $\mathbf{P}(\mathbf{X})$ at position $\mathbf{X}$, the locally-linear behaviour $\mathsf{C}(\mathbf{P}(\mathbf{X}))$ can be obtained (i.e. in practice, at the Gauss points used for the integration of the weak form). We thus have, using Voigt's notation,

$$\int_\Omega \mathbf{E}^*(\mathbf{X}) \cdot (\mathbf{C}(\mathbf{X})\mathbf{E}(\mathbf{X})) \, d\mathbf{X} = \int_{\Gamma_N} \mathbf{U}^*(\mathbf{X}) \cdot \mathbf{T}(\mathbf{X}) \, d\mathbf{X}. \tag{2.58}$$

where $\mathbf{T}(\mathbf{X})$ are the macro tractions acting on the Neuman boundary, Γ_N. This allows us, in turn, to obtain the macro displacement field ($\mathbf{U}(\mathbf{X})$) and from it, to update the strain and stress fields, to compute again the updated locally-linear behaviour. The process continues until convergence as mentioned in section 2.1.

2.3.2 Non-linear viscous fluids in porous media: manifold-based upscaling

Isothermal flows of non-linear fluids in complex microstructures can be simulated by solving the momentum and mass balance equations and a suitable rheological constitutive model. For inertialess incompressible flows, these balance equations read,

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \tag{2.59}$$

and

$$\nabla \cdot \mathbf{v} = 0. \tag{2.60}$$

Here, $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\mathbf{v}$ the velocity field, both defined at time t at each point within the fluid domain Ω_f. When considering porous media, the domain Ω is assumed fully saturated, with the fluid phase occupying the region Ω_f whereas the remaining part $\Omega_s = \Omega - \Omega_f$ is occupied by a solid phase assumed at rest.

An appropriate constitutive equation must be postulated to describe the fluid's rheology. There are many possible choices, the most usual ones being related to Newtonian, generalized-Newtonian and quasi-Newtonian fluids, as well as to suspensions, briefly summarized below:

- *Newtonian fluid.* For a Newtonian fluid, the constitutive equation reads

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} = -p\mathbf{I} + 2\eta\mathbf{D}, \tag{2.61}$$

  where p is the pressure field that can be interpreted as the Lagrange multiplier associated with the incompressibility constraint, $\mathbf{I}$ is the identity tensor, $\boldsymbol{\tau}$ the extra-stress tensor, η the constant fluid viscosity and $\mathbf{D}$ the rate of strain tensor, i.e. the symmetric part of the velocity gradient, $2\mathbf{D} = \nabla\mathbf{v} + (\nabla\mathbf{v})^T$.

- *Generalized-Newtonian fluid.* For a generalized-Newtonian fluid, the constitutive equation (2.61) remains formally unchanged, but a viscosity η that depends on the effective strain rate $\dot{\gamma}$. The latter is usually expressed from the second invariant of the rate of strain tensor, i.e. $\dot{\gamma} = \sqrt{2\mathbf{D}:\mathbf{D}}$. The simplest of such models is the power-law viscosity given by

$$\eta = \kappa\dot{\gamma}^{n-1}, \tag{2.62}$$

  where κ and n are the consistency and power-law index, respectively. The value $n = 1$ corresponds to a Newtonian fluid.

- *Quasi-Newtonian fluid.* In the quasi-Newtonian fluid model, the viscosity function is not merely a function of the second invariant of the rate of strain tensor but also depends on the relative rate of rotation of the fluid. Consequently, the quasi-Newtonian fluid is able to show e.g. shear-thinning in shear flow and extension-thickening in elongational flow. The constitutive equation for the quasi-Newtonian fluid reads

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta_{QN}\mathbf{D}, \tag{2.63}$$

where η_{QN} is the effective viscosity of the fluid that accounts for shear as well as extension according to the local type of flow. The latter is quantified by means of a scalar quantity χ that differentiates the type of regime (shear, elongation or rigid motion) [Ryssel & Brunn 1999, Thompson & Mendes 2005]. In [Ryssel & Brunn 1999] for example, the following viscosity function is proposed for 2D planar flows:

$$\eta_{QN}(\dot{\gamma}, \chi) = (\eta_S(\dot{\gamma}))^{f(\chi)} (\eta_E(\dot{\epsilon}))^{1-f(\chi)}, \tag{2.64}$$

with the shear viscosity η_S depending on $\dot{\gamma}$ (as in the case of generalized-Newtonian fluids) and the extensional viscosity η_E depending on $\dot{\epsilon}$, with $2\dot{\epsilon} = \dot{\gamma}$, and with the function $f(\chi)$ satisfying

$$f(\chi) = \begin{cases} 1, & \text{if } \chi = 1 \\ 0, & \text{if } \chi = 0 \end{cases}, \tag{2.65}$$

in order to recover the shear viscosity at locations exhibiting a shear flow and the extensional viscosity where planar extension occurs. We consider $f(1 < \chi \leq 2) = 1$ for approaching the zero shear rate viscosity in the limit case of rigid rotation.

- *Suspension of rigid rods in a Newtonian fluid.* When considering a population of rigid rods immersed into a Newtonian fluid, and making use of suitable simplifying assumptions [Binetruy *et al.* 2015], the constitutive equation for the suspension reads

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D} + 2\eta N_p(\mathbf{a} : \mathbf{D})\mathbf{a}, \tag{2.66}$$

where $\mathbf{a}$ is the so-called orientation tensor and N_p is a material parameter. The evolution of $\mathbf{a}$ is governed by the Folgar & Tucker model

$$\dot{\mathbf{a}} = \nabla\mathbf{v} \cdot \mathbf{a} - \mathbf{a} \cdot (\nabla\mathbf{v})^T - 2(\mathbf{a} : \mathbf{D})\mathbf{a} + \beta\left(\mathbf{a} - \frac{\mathbf{I}}{3}\right). \tag{2.67}$$

In view of the advective character of this equation, suitable boundary conditions must be specified for $\mathbf{a}$ at the inflow boundary of the RVE only, whereas velocity boundary conditions must be specified on the whole boundary.

2.3.2.1 Upscaling non-Newtonian fluids flowing in porous media

The flow model is solved in the representative volume $\omega(\mathbf{X})$, where two phases coexist, i.e. the fluid phase occupying the domain $\omega_f(\mathbf{X})$ and the solid phase, assumed rigid and at rest, occupying the region $\omega_s(\mathbf{X})$, with $\omega_f(\mathbf{X}) \cup \omega_s(\mathbf{X}) = \omega(\mathbf{X})$ and $\omega_f(\mathbf{X}) \cap \omega_s(\mathbf{X}) = \emptyset$. The flow model consists of the mass and momentum balance equations complemented by the constitutive equation discussed in the previous section, namely

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \\ \nabla \cdot \mathbf{v} = 0 \\ \boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} \end{cases}. \tag{2.68}$$

Here, $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{D})$ in the case of Newtonian, generalized-Newtonian and quasi-Newtonian fluids and $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{D}, \mathbf{a})$ in the case of rod suspensions.

The above governing equations are complemented with the boundary condition $\mathbf{v}(\mathbf{x} \in \partial\omega(\mathbf{X})) = \mathbf{V}$, where $\mathbf{V}$ comes from the macroscopic flow problem.

The solution of the flow problem (2.68) yields the velocity field $\mathbf{v}(\mathbf{x} \in \omega_f(\mathbf{X}))$, and from it the strain rate $\mathbf{D}(\mathbf{x} \in \omega_f(\mathbf{X}))$.

One can thus compute the power $\mathcal{DP}(\mathbf{V}; \mathbf{X})$ dissipated in the RVE and associated with the macroscopic velocity $\mathbf{V}$ prescribed at the boundary $\partial\omega$:

$$\mathcal{DP}(\mathbf{V}; \mathbf{X}) = \int_{\omega_f(\mathbf{X})} \boldsymbol{\sigma}(\mathbf{x}) : \mathbf{D}(\mathbf{x})d\mathbf{x}. \tag{2.69}$$

The specific microscopic dissipation $\mathcal{DP}^m$ is then obtained by dividing $\mathcal{DP}$ given by (2.69) by the RVE volume $|\omega(\mathbf{X})|$.

Obviously, the considered fluid models being purely viscous, they only involve dissipated power, and consequently the effective macroscopic model should account for it. In Darcy's model, the specific macroscopic dissipated power $\mathcal{DP}^M$ reads

$$\mathcal{DP}^M(\nabla P, \mathbf{V}) = \nabla P \cdot \mathbf{V}. \tag{2.70}$$

Thus, by equating the micro and macro-scale dissipations, we obtain

$$\mathcal{DP}^m = \nabla P|_{\mathbf{X}} \cdot \mathbf{V}(\mathbf{X}), \tag{2.71}$$

or by assuming the existence of an effective permeability $\mathbf{K}_{eff}(\mathbf{X})$,

$$\nabla P|_{\mathbf{X}} = \mathbf{K}_{eff}^{-1}(\mathbf{X})\mathbf{V}(\mathbf{X}), \tag{2.72}$$

from which we finally obtain our main result:

$$\mathcal{DP}^m(\mathbf{V}; \mathbf{X}) = \mathbf{K}_{eff}^{-1}(\mathbf{X}) : (\mathbf{V}(\mathbf{X}) \otimes \mathbf{V}(\mathbf{X})). \tag{2.73}$$

This expression constitutes a constructive definition of the effective permeability. For calculating the latter, it suffices to take the second derivative of $\mathcal{DP}^m(\mathbf{V})$ related to the microstructure existing at location $\mathbf{X}$:

$$\mathbf{K}_{eff}^{-1}(\mathbf{X}) = \frac{1}{2}\frac{d^2\mathcal{DP}^m(\mathbf{V}; \mathbf{X})}{d\mathbf{V}^2}. \tag{2.74}$$

We note that for a Newtonian fluid, the velocity, strain-rate and stress fields scale linearly with the velocity prescribed at the RVE boundary. Thus, the dissipated power scales with the square of the velocity, leading to a constant effective permeability [Lopez *et al.* 2015].

Thus, homogenized macro simulations can be performed very efficiently. After solving at a given iteration the macroscopic flow problem, from the computed velocity $\mathbf{V}$ at each macroscopic location $\mathbf{X}$, the associated permeability can be updated according to Eq. (2.74). This calculation is performed in almost real-time since the dissipative manifold has been precomputed offline. From the updated permeability, a new macroscopic calculation can be carried out.

2.3.3 Numerical examples

We consider two different microstructures, i.e. isotropic and orthotropic, as depicted in Fig. 2.23.
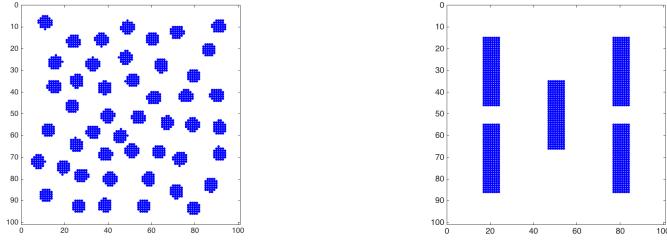
Figure 2.23 – RVE for two microstructures: isotropic (left) and orthotropic (right)
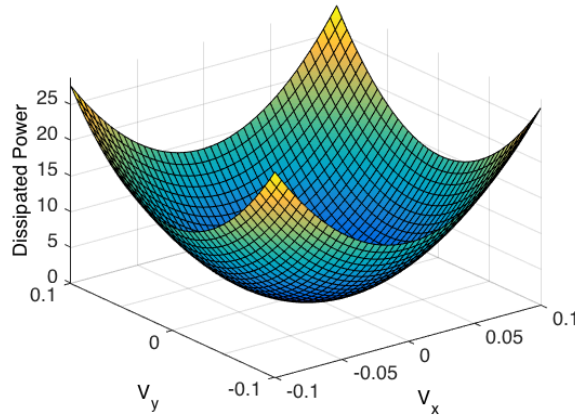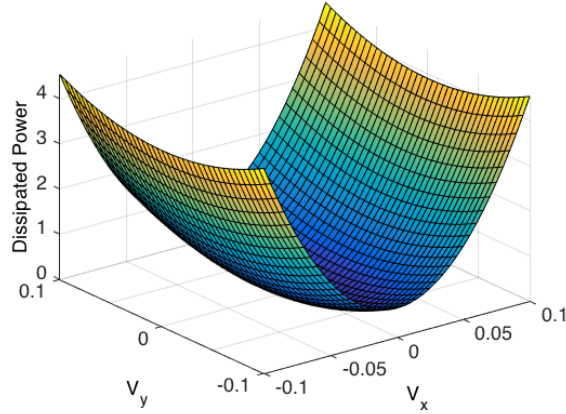

Figure 2.24 – Specific dissipated power for a Newtonian fluid flowing in the isotropic microstructure depicted in Fig. 2.23 (left)

2.3.3.1 Upscaling Newtonian fluids flowing in porous media

The specific dissipated power $\mathcal{DP}^M$ obtained for a Newtonian fluid is shown in Figs. (2.24) and (2.25) for the isotropic and anisotropic microstructures, respectively.

In view of (2.74), the components of the effective permeability tensor are obtained by taking the second derivatives of the dissipated power with respect to the velocity $\mathbf{V}$ prescribed at the RVE boundary, and inverting the resulting matrix. As discussed in [Lopez *et al.* 2015], the intrinsic permeability obtained for a Newtonian fluid is a purely geometrical property and can be obtained by multiplying the effective permeability by the viscosity.

Figures 2.26 and 2.27 depict the components of the effective permeability for a Newtonian fluid flowing in the isotropic and anisotropic microstructures shown in Fig. 2.23, left and right respectively. It can be noticed that the permeability does not depend on the prescribed velocity in view of the linear behaviour of the fluid. Moreover, for the almost isotropic microstructure, as expected, the diagonal components of the permeability tensor are quite similar, whereas the off-diagonal component is much smaller. For the orthotropic microstructure, a clear difference is noticed between the diagonal components, and the off-diagonal component vanishes. Indeed, the microstructure perfectly aligns with the coordinates axes that constitute the principal directions of the permeability tensor.

For the generalized-Newtonian and quasi-Newtonian fluid models, the results obtained using the proposed procedure based on the dissipated power manifold are, as expected, in

59

Figure 2.25 – Specific dissipated power for a Newtonian fluid flowing in the orthotropic microstructure depicted in Fig. 2.23 (right)

excellent agreement with the ones obtained using the analytical procedure described in our former works [Ammar *et al.* 2016, Lopez *et al.* 2016c].

2.3.3.2 Upscaling suspensions of rods flowing in porous media

As detailed in Section 3, the flow model for a suspension of rods consists of the momentum and mass balances complemented with the constitutive equation

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D} + 2\eta N_p(\mathbf{a} : \mathbf{D})\mathbf{a}, \tag{2.75}$$

that involves the conformation tensor $\mathbf{a}$ whose evolution is governed by the transport equation

$$\dot{\mathbf{a}} = \nabla\mathbf{v} \cdot \mathbf{a} - \mathbf{a} \cdot (\nabla\mathbf{v})^T - 2(\mathbf{a} : \mathbf{D})\mathbf{a} + \beta\left(\mathbf{a} - \frac{\mathbf{I}}{3}\right). \tag{2.76}$$

The resulting flow problem is solved in the RVE ω with the macroscopic velocity $\mathbf{V}$ specified at the RVE boundary $\partial\omega$. In order to solve the orientation equation (2.76), one must specify the orientation tensor at the inflow boundary $\partial^-\omega \subset \partial\omega$ characterized by $\mathbf{V} \cdot \mathbf{n}(\mathbf{x} \in \partial\omega) < 0$, where $\mathbf{n}$ is the outward unit vector normal to $\partial\omega$ at point $\mathbf{x}$.

The need for these inflow conditions is a really difficult issue. Indeed, while we can naturally define at the macroscopic scale an upscaled orientation tensor $\mathbf{A} \equiv \langle\mathbf{a}\rangle$, we cannot formulate a proper evolution equation for $\mathbf{A}$ at the macroscopic scale in terms of the averaged velocities $\mathbf{V}$ appearing in the Darcy model. The orientation evolution is induced and driven by the local velocity gradients existing at the microscopic scale. It is assumed that the characteristic fibers length is lower than the characteristic channels diameter in order to neglect confinement effects addressed in [Perez *et al.* 2016, Scheuer *et al.* 2016].

When solving the micro-scale flow problem, the orientation to be prescribed at the inflow boundary $\partial^-\omega$ is in fact undetermined. In order to quantify the impact of this indeterminacy, we decided to specify arbitrary orientation states at $\partial^-\omega$. In view of the intense tortuosity that complex microstructures entail, we found that the orientation field rapidly forgets its entrance condition (fading memory) and is essentially velocity-driven. Thus, the micro-scale solution depends almost exclusively on the prescribed velocity $\mathbf{V}$ that arises from the macroscopic scale. This empirical observation is of crucial importance.
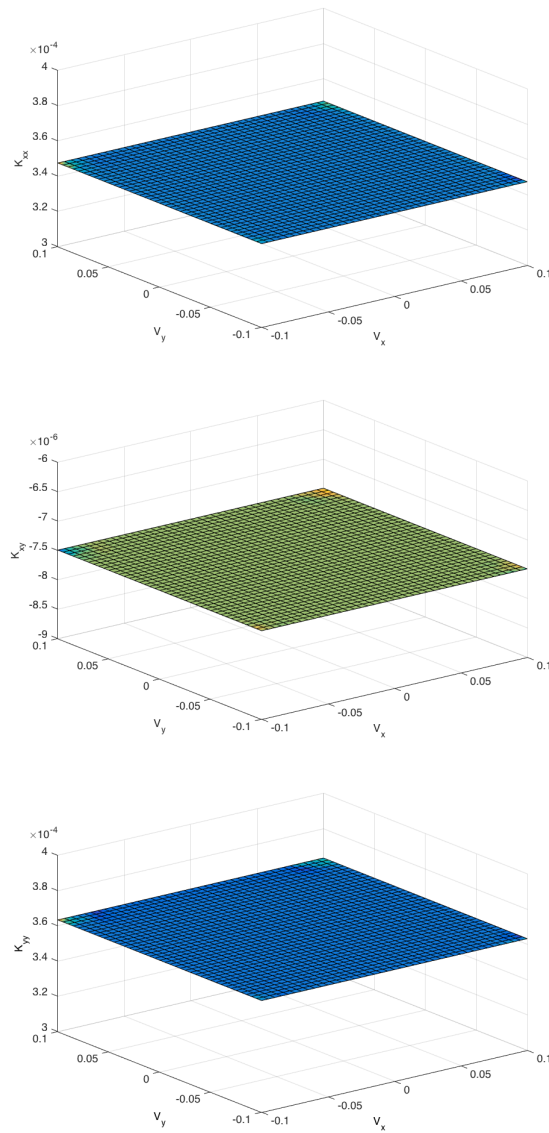
Figure 2.26 – Newtonian fluid and isotropic microstructure. Components of the effective permeability tensor: K_{xx} (top), K_{xy} (middle) and K_{yy} (bottom)
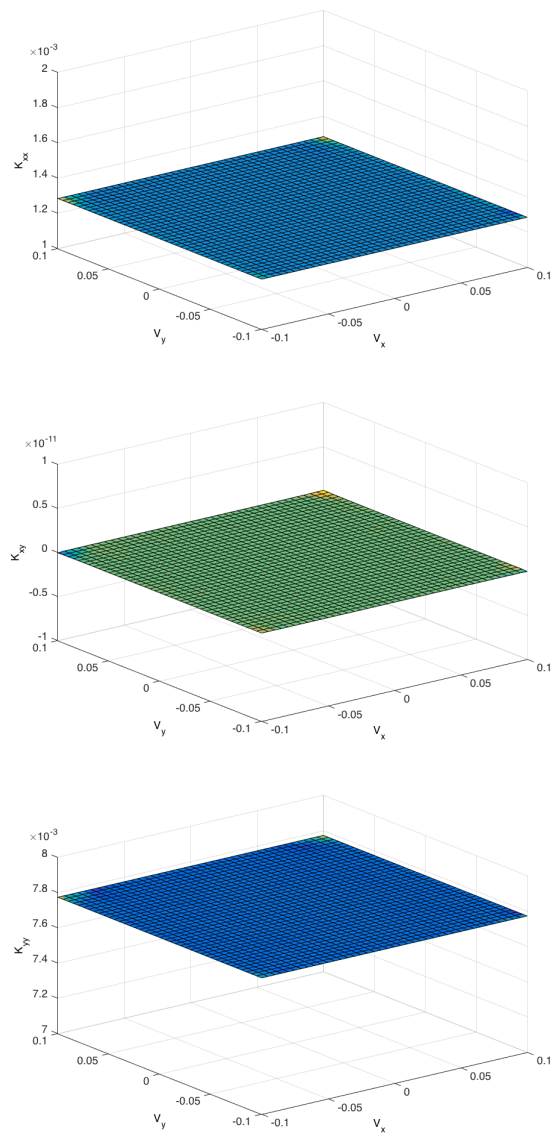
Figure 2.27 – Newtonian fluid and orthotropic microstructure. Components of the effective permeability tensor: K_{xx} (top), K_{xy} (middle) and K_{yy} (bottom)

In practice, any orientation can thus be prescribed at the inflow boundary $\partial^-\omega$, the simplest ones being: (i) an isotropic orientation state $\mathbf{a}(\mathbf{x} \in \partial^-\omega) = \mathbf{I}/2$ or $\mathbf{I}/3$ in the 2D or 3D cases, respectively; (ii) the local alignement of rods with the incoming flow, $\mathbf{a}(\mathbf{x} \in \partial^-\omega) = (\mathbf{V} \otimes \mathbf{V})/\|\mathbf{V}\|^2$; or the macroscopic orientation existing at that position at the previous iteration, $\mathbf{a}(\mathbf{x} \in \partial^-\omega) = \hat{\mathbf{A}}$. Thus, with the macroscopic velocity $\mathbf{V}$ specified on the whole boundary $\partial\omega$ and the orientation at the inflow boundary $\partial^-\omega$, the micro-scale flow problem is solved to obtain $\mathbf{v}(\mathbf{x})$, $\mathbf{D}(\mathbf{x})$, $\boldsymbol{\tau}(\mathbf{x})$ and $\mathbf{a}(\mathbf{x})$ at each position $\mathbf{x}$ in the fluid domain ω_f, and from those, the dissipated power $\mathcal{DP}$ and the averaged orientation $\mathbf{A} = \langle \mathbf{a}(\mathbf{x}) \rangle$.

Since the outputs of the microscopic calculation are almost entirely velocity-driven, we can assume the existence of the two manifolds $\mathcal{DP}(\mathbf{V})$ and $\mathbf{A}(\mathbf{V})$. The first manifold yields the effective permeability, while the second gives a macroscopic descriptor of the orientation field.

Figure 2.28 compares the components of the macroscopic orientation tensor $\mathbf{A}$ as a function of the prescribed velocity $\mathbf{V}$, obtained for the isotropic and orthotropic microstructures of Fig. 2.23. The expected symmetries are noticed for the isotropic microstructure, while a preferential orientation along the y-direction is predicted in the orthotropic case.

Unlike for a Newtonian fluid, the effective permeability of the suspension does depend on the prescribed velocity $\mathbf{V}$, in view of the non-linearity of the micro-scale problem. This can be seen in the results shown in Fig. 2.29, obtained for the isotropic microstructure. Comparison with the Newtonian results shows that the presence of rods yields an overall decrease of the effective permeability.

2.3.4   Conclusions

This section explored the possibility of performing manifold-based upscaling of linear and non-linear models for flows in porous media. We proposed the construction of the so-called dissipated power manifold whose second derivative with respect to the macroscopic velocity (enforced at the boundary of the RVE) yields the inverse of the effective permeability tensor. When solving the micro-scale problem parametrically by means of the PGD, construction of the dissipated power manifold is straightforward and extremely fast from the computational point of view.

The second contribution of the present work is the consideration of suspensions of rods flowing in porous media. Through numerical experiments, we found that the dissipated power is insensitive to the orientation state that must be specified at the inflow boundary of the RVE, and thus mostly depends on the macroscopic velocity prescribed at the RVE boundary. This crucial observation allowed us to apply the proposed upscaling procedure to complex fluids endowed with a fading memory.
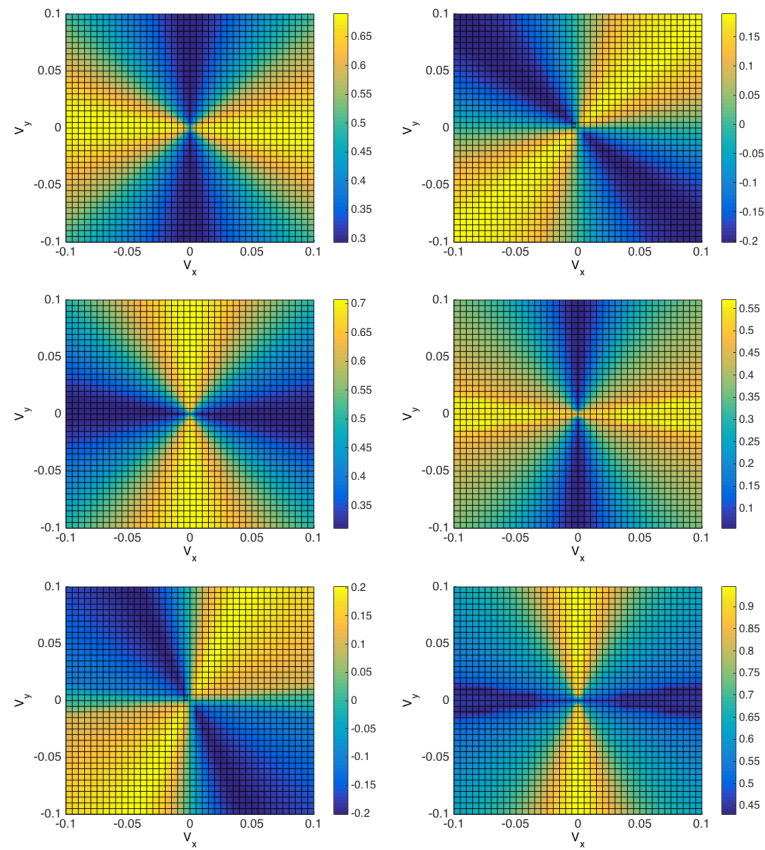
Figure 2.28 – Components of the upscaled orientation tensor $\mathbf{A}$ (A_{xx} in the left, A_{xy} in the center and A_{yy} in the right) for the isotropic (top) and orthotropic (bottom) microstructures (the colour-bar is different in each graph)
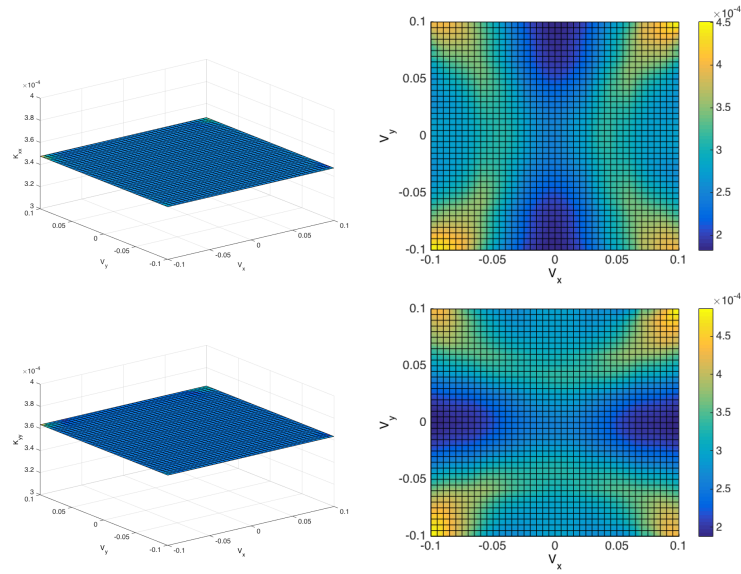
Figure 2.29 – Isotropic microstructure: comparing the effective permeability components K_{xx} (top) and K_{yy} (bottom) for a Newtonian fluid (left) and a suspension of rods (right)

Data Completion techniques

Abstract   Model identification of underlying physical laws by means of data is specially cumbersome if the sought dynamics live in a high dimensional space. This usually involves the need for large amount of data, unfeasible in such a high dimensional settings. Robust data completion methodologies which are able to operate in such circumstances will be of crucial interest within the Dynamic Data Driven Application Systems framework. These data completion techniques seems a seamless bridge to exploit the information carried by each data point, thus, constructing efficient response surfaces, which will be ultimately used by data driven algorithms.

Contents

This chapter presents an overview of different methodologies to generate response surface based on scattered data. These methodologies are of crucial importance within the Data-Driven framework whenever the data has to be interpolated/extrapolated to infer behaviours at locations where there are no experimental measurements. Section 3.1 presents a methology called sparse Proper Generalized Decomposition (sPGD), which combines PGD [Cueto *et al.* 2016] rationale together with a collocation procedure. Indeed, this methology alleviates the so-called curse of dimensionality by means of the separation of variables and it makes use of the data only at the experimental sample points by virtue of the collocation method.

Section 3.2 provides three industrial applications where the sPGD methodology has been succesfully applied. The first application involves design and geometrical optimization of a vehicle structure to meet safety requirements while a crash test. The second application consists of identifiying the plastic behaviour of a given material, whereas the third application links the geometrical parameters of real surfaces with its performance in an automated tape placement (ATP) process.

Finally, section 3.3 exports the use a novel methodology called compressed sensing, which has been widely applied in the image processing field. This technique claims to reproduce an image, up to a certain accuracy, not using every single pixel than constitutes the image but only few of them. This procedure has been applied within the computational mechanics framework in [Brunton *et al.* 2016] to discover the underlying physics behind a dataset. In our work we extend the methodology to other scenarios such as transient problems or parametric models, where response surfaces are generated only by knowing the information at certain locations.

These topics are addressed in five papers, three of them already published and two submitted, that constitute this chapter:

- R. Ibáñez, E. Abisset-Chavanne, A. Ammar, D. Gonzalez, E. Cueto, A. Huerta, J. L. Duval, F. Chinesta, *A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition.*Complexity, **12**, 1-11, 2018.

- V. Limousin, X. Delgerie, E. Leroy, R. Ibáñez, C. Argerich, F. Daim, J. L. Duval, F. Chinesta, *Model Order Reduction and Machine Learning Techniques in Advanced Structural Mechanics. Application to Crash Simulation.* To be submitted into Mechanics & Industry.

- R. Ibáñez, E. Abisset-Chavanne, D. Gonzalez, J.L. Duval, E. Cueto, F. Chinesta, *Hybrid constitutive modeling: data-driven learning of corrections to plasticity models.* International Journal of Material Forming, DOI:10.1007/s12289-018-1448-x, 1-9, 2018.

- C. Argerich, A. Leon, R. Ibáñez, A. Barasinski, E. Abisset-Chavanne, F. Chinesta, *Prediction of composites behavior undergoing an ATP process through data-mining.* AIP Conference Proceedings, DOI: 10.1063/1.5034817, 2018.

- R. Ibáñez, E. Abisset-Chavanne, A. Ammar, E. Cueto, J.L. Duval, F. Chinesta, *Some applications of compressed sensing in computational mechanics.* Submitted to Computational Mechanics.

3.1 A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition

In what follows we present a methodology that combines the concept of separate representations to overcome the curse of dimensionality together with a collocation procedure to use data wherever is available. Such separate representation has previously been employed by the authors to construct a priori reduced-order modeling techniques, coined as Proper Generalized Decompositions [Chinesta *et al.* 2010] [Chinesta *et al.* 2014] [Chinesta & Ladeveze 2014] [Gonzalez *et al.* 2010] [Cueto *et al.* 2016] [Badías *et al.* 2017] [Gonzalez *et al.* 2017]. This will give rise to a sparse Proper Generalized Decomposition (sPGD in what follows) approach to the problem. Subsection 3.1.1 explains the basics of the technique. Subsection 3.1.2 shows two ingredients which will enhance the stability of the method and ultimately its power to predict new query points. We then analyze the performance of the just developed technique through a series of academic numerical experiments in subsection 3.1.3. Examples in up to ten dimensions are introduces, showing the robustness of the method for relatively high dimensional spaces.

3.1.1 Basics of the technique

For the ease of the exposition and, above all, representation, but without loss of generality, let us begin by assuming that the unknown objective function $f(x, y)$ lives in $\mathbb{R}^2$ and that is to be recovered from sparse data. As in previous references, see for instance [Mangan *et al.* 2016], we have chosen to begin with a Galerkin projection, in the form

$$\int_\Omega w^*(x, y)\left(u(x, y) - f(x, y)\right) dx dy = 0, \tag{3.1}$$

where $\Omega \subset \mathbb{R}^2$ stands for the—here, still two-dimensional—domain in which the identification is performed and $w^*(x, y) \in \mathcal{C}^0(\Omega)$ is an arbitrary test function. Finally, $u(x, y)$ will be the obtained approximation to $f(x, y)$, still to be constructed. In previous works of the authors [González *et al.* 2018b] as well as in other approaches to the problem (e.g., [Mangan *et al.* 2016]), this projection is subject to additional constraints of thermodynamic nature. In this work no particular assumption is made in this regard, although additional constraints could be imposed to the minimization problem.

Following the same rationale behind the Proper Generalized Decomposition (PGD), the next step is to express the approximated function $u^M(x, y) \approx u(x, y)$ as a set of separate one-dimensional functions,

$$u^M(x, y) = \sum_{k=1}^{M} X^k(x) Y^k(y). \tag{3.2}$$

The determination of the precise form of functional pairs $X^k(x)Y^k(y)$, $k = 1, \ldots, M$, is done by first projecting them on a finite element basis and by employing a greedy algorithm such that, once the approximation up to order $M - 1$ is known, the new M-th order term

$$u^M(x, y) = u^{M-1}(x, y) + X^M(x)Y^M(y) = \sum_{k=1}^{M-1} X^k(x)Y^k(y) + X^M(x)Y^M(y)$$

is found by any non-linear solver (Picard, Newton, ...)

It is well-known that this approach produces optimal results for elliptic operators (here, note that we have in fact an identity operator acting on u) in two dimensions, see [Chinesta *et al.* 2010] and references therein. There is no proof, however, that this separate representation will produce optimal results (in other words, will obtain *parsimonious models*) in dimensions higher than two. In two dimensions and with $w^* = u^*$ it provides the singular value decomposition of $f(x, y)$ [Chinesta *et al.* 2014]. Our experience, nevertheless, is that it produces almost optimal results in the vast majority of the problems tested so far.

It is worth noting that the product of the test function $w^*(x, y)$ times the objective function $f(x, y)$ is only evaluated at few locations (the ones corresponding to the experimental measurements) and that, in a general high dimensional setting, we will be in the low-data limit necessarily. Several options can be adopted in this scenario. For instance, the objective function can be first interpolated in the high dimensional space (still 2D in this introductory example) and then integrated together with the test function. Indeed, this will be the so-called *PGD in approximation* [Chinesta *et al.* 2014], commonly used when either $f(x, y)$ is known everywhere and a separated representation is sought or if $f(x, y)$ is known in a separated format but a few pairs M are needed for any reason. Under this rationale the converged solution $u(x, y)$ tries to capture the already interpolated solution in the high dimensional space but in a more compact format. As a consequence, the error due to interpolation of experimental measurements on the high dimensional space will persist in the final separate identified function.

In order to overcome such difficulties, we envisage a projection followed by interpolation method. However since information is just known at P sampling points (x_i, y_i), $i = 1, \ldots, P$, it seems reasonable to express the test function not in a finite element context, but to express it as a set of Dirac delta functions collocated at the sampling points,

$$\begin{aligned} w^*(x, y) =& u^*(x, y) \sum_{i=1}^{P} \delta(x_i, y_i) \\ =& \left(X^*(x)Y^M(y) + X^M(x)Y^*(y) \right) \sum_{i=1}^{P} \delta(x_i, y_i), \end{aligned} \tag{3.3}$$

giving rise to

$$\begin{aligned} & \int_\Omega w^*(x, y) \left(u(x, y) - f(x, y) \right) dxdy \\ =& \int_\Omega u^*(x, y) \sum_{i=1}^{P} \delta(x_i, y_i) \left(u(x, y) - f(x, y) \right) dxdy = 0, \end{aligned}$$

The choice of the test function $w^*(x, y)$ in the form dictated by Eq. (3.3) is motivated by the desire of employing a collocation approach while maintaning the symmetry of standard Bubnov-Galerkin projection operation.

Let us detail now the finite element projection of the one-dimensional functions $X^k(x)$, $Y^k(y)$, $k = 1, \ldots, M$, (often referred to as *modes*) appearing in Eq. (3.2). Several options

can be adopted, ranging from standard piecewise linear shape functions, global non-linear shape functions, maximum entropy interpolants, splines, kriging, etc. Regarding the kind of interpolant to use, an analysis will be performed in the sequel. Nevertheless, no matter which precise interpolant is employed, it can be expressed in matrix form as

$$X^k(x) = \sum_{j=1}^{N} N_j^k(x)\alpha_j^k = \begin{bmatrix} N_1^k(x) \dots N_N^k(x) \end{bmatrix} \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} = (\mathbf{N}_x^k)^T \mathbf{a}^k, \tag{3.4}$$

$$Y^k(y) = \sum_{j=1}^{N} N_j^k(y)\beta_j^k = \begin{bmatrix} N_1^k(y) \dots N_N^k(y) \end{bmatrix} \begin{bmatrix} \beta_1^k \\ \vdots \\ \beta_N^k \end{bmatrix} = (\mathbf{N}_y^k)^T \mathbf{b}^k, \tag{3.5}$$

where α_j^k and β_j^k, $j = 1, \dots, N$, represent the degrees of freedom of the chosen approximation. We employ $\mathbf{N^k}$ as the most usual nomenclature for the shape function vector. It is important to remark that the approximation basis could even change from mode to mode (i.e., for each i). For the sake of simplicity we take the same number of terms for both $X^k(x)$ and $Y^k(y)$, namely, N.

By combining Eqs. (3.1)-(3.5) a non linear system of equations is derived, due to products of terms in both spatial directions. An alternate direction scheme is here preferred to linearize the problem, which is also a typical choice in the PGD literature. Note that, when computing modes $X^M(x)$, the variation in the other spatial direction vanishes, $Y^*(y) = 0$, and vice versa.

In order to fully detail the matrix form of the resulting problem, we first employ the notation "$\otimes$" as the standard tensorial product (i.e., $\mathbf{b} \otimes \mathbf{c} = b_i c_j$), and define the following matrices

$$\mathbf{A}_x^{k\ell} = \mathbf{N}_x^k \otimes \mathbf{N}_x^\ell,$$

$$\mathbf{A}_y^{k\ell} = \mathbf{N}_y^k \otimes \mathbf{N}_y^\ell,$$

$$\mathbf{C}_{xy}^{k\ell} = \mathbf{N}_x^k \otimes \mathbf{N}_y^\ell.$$

For the sake of simplicity but without loss of generality, evaluations of the former operators at point (x_i, y_i) are denoted as

$$\mathbf{A}_{x_i}^{k\ell} = \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_x^\ell(x_i),$$

$$\mathbf{A}_{y_i}^{k\ell} = \mathbf{N}_y^k(y_i) \otimes \mathbf{N}_y^\ell(y_i),$$

$$\mathbf{C}_{x_i y_i}^{k\ell} = \mathbf{N}_x^k(x_i) \otimes \mathbf{N}_y^j(y_i).$$

Eqs. (3.6)-(3.7) below show the discretized version of the terms appearing in the weak form, Eq. (
refsec3-1:weakform1), when computing modes in the x direction. Again, M stands for the number of modes in the solution $u(x, y)$ while P denotes the number of sampling points.

$$\int_\Omega u^*(x, y) \sum_{i=1}^{P} \delta(x_i, y_i) u(x, y) dx dy$$

$$= \sum_{k=1}^{M} \sum_{i=1}^{P} \left((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k \right) \left((\mathbf{a}^*)^T \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k \right), \tag{3.6}$$

$$\int_{\Omega} u^*(x,y) \sum_{i=1}^{P} \delta(x_i, y_i) f(x,y) dx dy = \sum_{i=1}^{P} f(x_i, y_i) \left((\mathbf{a}^*)^T \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M \right). \tag{3.7}$$

Hence, by defining

$$\mathbf{M}_x = \sum_{i=1}^{P} ((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{MM} \mathbf{b}^M) \mathbf{A}_{x_i}^{MM},$$

$$\mathbf{m}_x = \sum_{k=1}^{M-1} \sum_{i=1}^{P} ((\mathbf{b}^M)^T \mathbf{A}_{y_i}^{Mk} \mathbf{b}^k) \mathbf{A}_{x_i}^{Mk} \mathbf{a}^k,$$

$$\mathbf{f}_x = \sum_{i=1}^{P} f(x_i, y_i) \mathbf{C}_{x_i y_i}^{MM} \mathbf{b}^M,$$

allows to write a system of algebraic equations

$$\mathbf{M}_x \mathbf{a}^M = \mathbf{f}_x - \mathbf{m}_x. \tag{3.8}$$

Exactly the same procedure is followed to obtain an algebraic system of equations for $\mathbf{b}^M$. This allows to perform an alternating directions scheme to extract a new couple of $X^M(x)$ and $Y^M(y)$ modes.

This formulation has several aspects that deserve to be highlighted:

1. No assumption about $f(x,y)$ has been made other than assuming known its value at sampling points. Indeed, both problems of either interpolating or making a triangulation in a high dimensional space are circumvented due to the separation of variables.

2. The operator $\mathbf{M}_x$ is composed of P rank-one updates. Meaning that the rank of such operator is at most P. Furthermore, if a subset of measured points share the same coordinate x_i, the entire subset will increase the rank of the operator in one unity.

3. The position of the sampling points will constraint the rank of the PGD operators. That is the reason why, even if the possibility of having a random sampling of points is available, it is always convenient to perform a smart sampling technique such that the rank in each direction tends to be maximized. Indeed, the higher the rank of the PGD operator is, the more cardinality of $\mathbf{a}$ and $\mathbf{b}$ can be demanded without degenerating into an underdetermined system of equations.

There are plenty of strategies to smartly select the position of the sampling points. They are based on either knowing an a priori error indicator or having a reasonable estimation of the sought response surface. Certainly, an adaptive strategy based on the gradient of the precomputed modes could be envisaged. However, the position of the new sampling points will depend on the response surface calculated using the previous sampling points, making parallelization difficult. That is the reason why latin hypercube is chosen in the present work. Particularly, latin hypercube tries to collocate P sampling points in such a way that the projection of those points into x and y axis are as far as possible.

3.1.2 Enhancing stability and prediction

Even though the proposed methodology generates a response surface that minimize the distance to the sampling points, nothing has been said in the subsection 3.3 with respect to the inference capabilities outside the sampling points. This subsection discusses the importance of selecting properly the interpolant function for the 1D basis just like the

possibility of changing the interpolant basis as the number of modes (or enrichements)
increases. Indeed, both strategies can be seen as smoothers outside the sampling points,
avoiding spurious oscillations out of the sample points.

3.1.2.1   Choice of the 1D basis

Up to now, nothing has been specified about the basis in which each one of the one-
dimensional modes was expressed. Despite of the fact that selecting the best 1D basis is
problem dependent i.e. depends on the topology of the response surface to be captured,
there are some basis which tend to smooth the solution outside the control points. One may
think of smooth global polynomial functions like Chebychev polynomials among others. In
this subsection, we will introduce an interpolant based on Kriging techniques. Simple Krig-
ing has been used throughout history in order to get relatively smooth solutions, avoiding
spurious oscillations characteristic of high order polynomial interpolation. This phenomena
is called Runge's phenomenon. It appears due to the fact that the sampling point locations
are not chosen properly, i.e., they will not be collocated, in general, at the Gauss-Lobato-
Chebychev quadrature points. Kriging interpolants consider each point as a realization of
a Gaussian process, so that high oscillations are considered as unlikely events.

Hence, by defining a spatial correlation function based on the relative distance between
two points, $\mathcal{D}(x_i - x_j) = \mathcal{D}_{ij}$, an interpolant is created over the separated 1D domain,

$$X^k(x) = \sum_{i=1}^{N} \frac{\alpha_i^k}{N} + \sum_{j=1}^{N} \lambda(x - x_j) \left(\alpha_j^k - \sum_{l=1}^{N} \frac{\alpha_l^k}{N} \right),$$

where $\lambda(x - x_j)$ is a weighting function which strongly depends on the definition of the
correlation function, and the α_i coefficients are the nodal values associated to the x_i Kriging
control points. Note that these control points are not the sampling points. We have
chosen this strategy so as to allow us to accomplish an adaptivity strategy that will be
described next. In the present work, these control points are uniformly distributed along
the 1D domain. Although several definitions of the correlation function exist, a Gaussian
distribution is chosen as

$$\mathcal{D}_{ij} = \mathcal{D}(x_i - x_j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - x_j)^2}{2\sigma^2}},$$

where σ is the variance of the Gaussian distribution. Several a priori choices can be adopted
to select the value of the variance based on the distance between two consecutive control
points, e.g., $\sigma = h\sqrt{(x_{i+1} - x_i)^2}$. The magnitude of h should be adapted depending on the
desired global character of the support. To ensure the positivity of the variance, h should
be in the interval $]0, +\infty[$.

Let us define now a set of C control points

$$\mathbf{x}^{cp} = [x_1^{cp}, x_2^{cp}, \ldots, x_C^{cp}],$$

and the P sampling points

$$\mathbf{x}^{sp} = [x_1^{sp}, x_2^{sp}, \ldots, x_P^{sp}].$$

Let us define in turn a correlation matrix between all control points and a correlation matrix
between the control points and the sampling points as

$$\mathcal{C}_{ij}^{cp-cp} = \mathcal{D}(x_i^{cp} - x_j^{cp}),$$

$$\mathcal{C}_{ij}^{cp-sp} = \mathcal{D}(x_i^{cp} - x_j^{sp}).$$
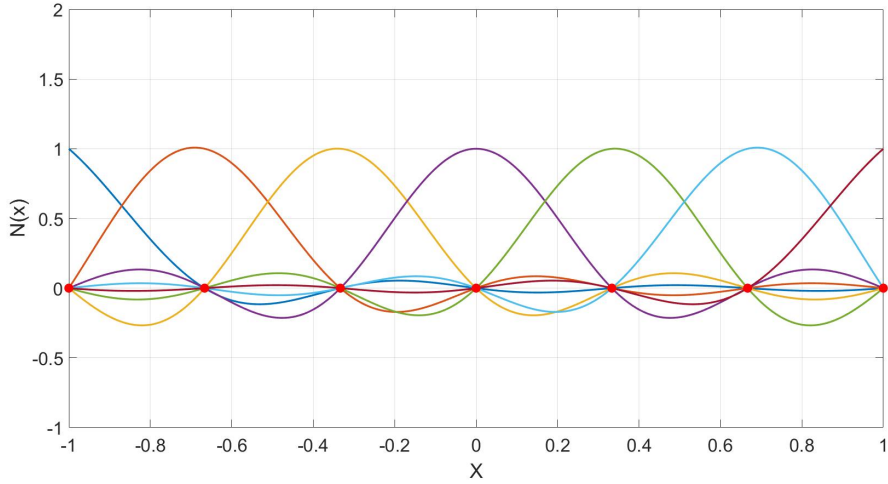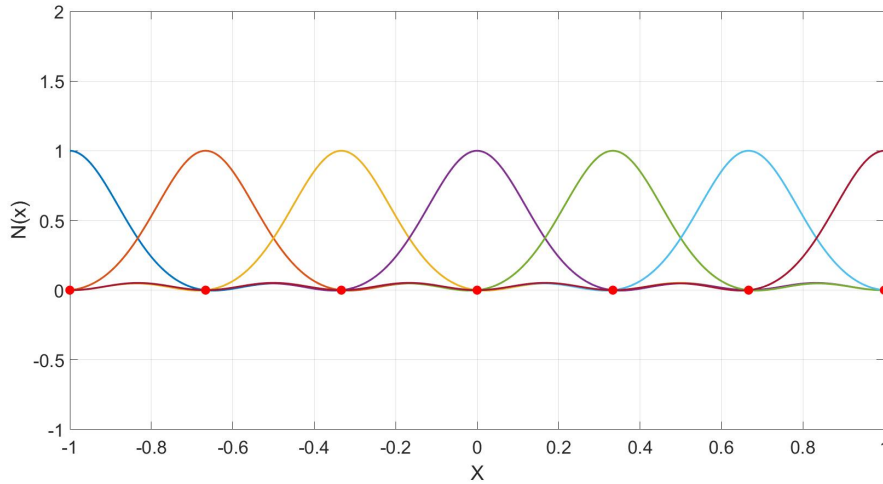
Figure 3.1 – Kriging shape functions using $\sigma = \sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

Under these settings, we define a weighting function for each control point and for each sampling point as

$$\mathbf{\Lambda} = (\boldsymbol{\mathcal{C}}^{cp-cp})^{-1} \boldsymbol{\mathcal{C}}^{cp-sp},$$

where $\lambda(x_i^{cp} - x_j^{sp}) = \Lambda_{ij}$.

If we reorganize the terms in the same way that we did in the previous section to have a compact and close format of the shape function $\mathbf{N}_x^k$, we arrive to

$$X^k(x_j^{sp}) = \sum_{i=1}^{N} N_i^k(x_j^{sp})\alpha_i^k = [N_1^k(x_j^{sp}) \dots N_N^k(x_j^{sp})] \begin{bmatrix} \alpha_1^k \\ \vdots \\ \alpha_N^k \end{bmatrix} = (\mathbf{N}_{x_j^{sp}}^k)^T \mathbf{a}^k,$$

where each shape function is given by:

$$N_i^k(x_j^{sp}) = \frac{1 - \sum_{j=1}^{N} \Lambda_{ij}}{N} + \Lambda_{ij}.$$

Figs. 3.1-3.2 depict the appearance of the simple Kriging interpolants using 7 control points uniformly distributed along the domain, for $h = 1$ and $h = \frac{1}{3}$, respectively. It can be highlighted that both the Kronecker delta (i.e., strict interpolation) and partition of unity properties are satisfied for any value of h. Moreover, it is worth noting that the higher the variance the correlation function has, the more global the shape functions are. Furthermore, it is known that 99 per cent of the probability of a Gaussian distribution is comprised within a interval of $[m - 3\sigma, m + 3\sigma]$, being m the mean value of the distribution. This issue explains perfectly well why the support of each Gaussian distribution takes 2 elements for the case where $h = \frac{1}{3}$. Indeed, the shape of the interpolants is quite similar to standard finite element shape functions, but with a Gaussian profile. The remaining 1 per cent of probability is comprised in the small ridges happening in the middle of the elements.

In light of these results, a family of interpolants based on Kriging can be easily created just selecting the value of the variance within the correlation function. Therefore, globality of the support can be easily adjusted always under the framework of the partition of unity.

Figure 3.2 – Kriging shape functions using $\sigma = \frac{1}{3}\sqrt{(x_{i+1} - x_i)^2}$ for 7 control points uniformly distributed along the 1D domain.

3.1.2.2 Modal adaptivity strategy

In a standard PGD framework, the final solution is approximated as a sum of M modes or functional products, see eq. (3.2). Each one of the separated modes must be projected onto a chosen basis to render the problem finite dimensional. A standard choice is to select the same basis for each one of the modes:

$$\mathbf{N}^1 = \mathbf{N}^2 = \ldots = \mathbf{N}^M.$$

Despite of the fact that this choice seems reasonable, when dealing with non-structured sparse data, it may not be such. In the subsection 3.1.1 we proved that the rank of the separated system strongly depends on the distribution of the data sampling. Therefore, the cardinality of the interpolation basis must not exceed the maximum rank provided by the data sampling. Indeed, this constraint, which provides an upper bound to build the interpolation basis, only guarantees that the minimization is satisfied at the sampling points, without saying anything out of the measured points. Hence, if sampling points are not abundant, in the limit of low-data regime, high oscillations may appear out of these measured points. These oscillations are not desirable since the resulting prediction properties could be potentially decimated.

In order to tackle this problem, we take advantage of the residual-based nature of the PGD. Indeed, the greedy PGD algorithm tries to enrich a solution composed by M modes,

$$u^M(x, y) = \sum_{k=1}^{M} X^k(x) Y^k(y),$$

just by looking at the residual that accounts for the contribution of the previous modes, as shown in Eq. (3.8).

Therefore, an appealing strategy to minimize spurious oscillations out of the sampling points is to start the PGD algorithm looking for modes with relatively smooth basis (for instance, Kriging interpolants with a few control points). Therefore, an indicator in order

to make an on-line modal adaptive strategy is required. In the present work, we use the norm of the PGD residual,

$$\mathcal{R}_{\mathcal{P}}^M = \frac{1}{\sqrt{P}} \sqrt{\sum_{i \in \mathcal{P}} \left(f(x_i, y_i) - u^M(x_i, y_i) \right)^2},$$

where $\mathcal{P}$ is the set of P measured points and $f(x, y)$ is the function to be captured.

In essence, when the residual norm stagnates, a interpolant basis is enriched, i.e. introducing one more control point and always uniformly distributed in the case of Kriging interpolants, following

$$\Delta \mathcal{R}_{\mathcal{P}}^M = \mathcal{R}_{\mathcal{P}}^M - \mathcal{R}_{\mathcal{P}}^{M-1} < \epsilon_r.$$

By doing this, oscillations are reduced, since higher-order basis will try to capture only what remains in the residual. Here, ϵ_r is a tolerance defining the resilience of the sPGD to increase the cardinality of the interpolation basis. The lower ϵ_r is, the more resilient the method is to increase the cardinality.

To better understand the method, we will quantify the error for two set of points: the first set is associated to the sampling points, $\mathcal{P}$,

$$\mathcal{E}_{\mathcal{P}} = \frac{1}{\#\mathcal{P}} \sum_{s \in \mathcal{P}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}},$$

where $f(x_s, y_s)$ is assumed not to vanish and where $\mathcal{L}$ also includes points other than the sampling points. This is done in order to validate the algorithm, by evaluating the reference solution—which is a priori unknown in a general setting—at points different to the sampling ones,

$$\mathcal{E}_{\mathcal{L}} = \frac{1}{\#\mathcal{L}} \sum_{s \in \mathcal{L}} \sqrt{\frac{(f(x_s, y_s) - u^M(x_s, y_s))^2}{f(x_s, y_s)^2}}.$$

Since the s-PGD algorithm minimizes the error only at the sampling points $\mathcal{P}$ it is reasonable to expect that $\mathcal{E}_{\mathcal{P}} \leq \mathcal{E}_{\mathcal{L}}$.

To test the convergence of the just presented algorithm, we consider

$$f_1(x, y) = (\cos(3\pi x) + \sin(3\pi y))y^2 + 4,$$

that presents a quite oscillating behavior along the x direction, whereas the y direction is quadratic. We are interested in capturing such a function in the domain $\Omega_y = \Omega_x = [-1, 1]$.

Figs. (3.3)-(3.4) show the errors $\mathcal{E}_{\mathcal{P}}$ and $\mathcal{E}_{\mathcal{L}}$ in identifying the function $f_1(x, y)$. In this case, we consider two distinct possibilities: no modal adaptivity at all, and a modal adaptivity based on the residual, respectively. Several aspects can be highlighted. The first one is that $\mathcal{E}_{\mathcal{P}}$ (asterisks) decreases much faster when there is no modal adaptivity. This is expected, since we are minimizing with a richer basis since the very beginning, instead of starting with smooth functions like in the residual based approach. However, even if the minimization in the sampling points is well achieved, when no modal adaptivity is considered, the error out of the sampling points may increase as the solution is enriched with new modes. Nevertheless, the residual-based modal adaptivity alleviates this problem. As it can be noticed, starting with relatively smooth functions drives the solution out of the sampling points to be smooth as well, avoiding the problem of high oscillations appearing out of the sampling points.
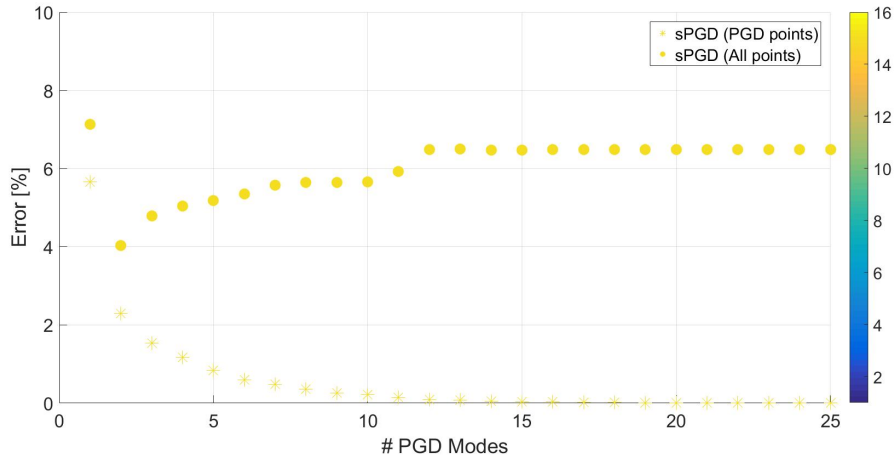
Figure 3.3 – $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. No modal adaptivity.
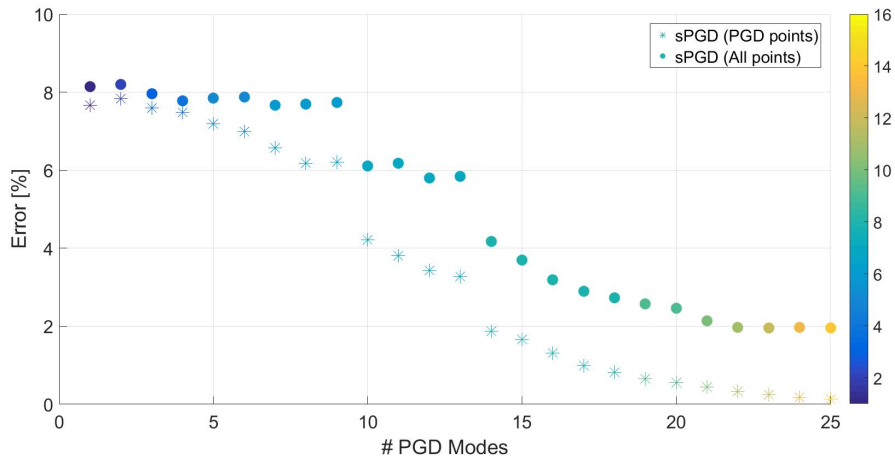
Figure 3.4 – $\mathcal{E}_{\mathcal{L}}$ (points) and $\mathcal{E}_{\mathcal{P}}$ (asterisk) versus the number of modes for $f_1(x, y)$, $\#\mathcal{P} = 100$, $\#\mathcal{L} = 1000$. Modal adaptivity based on the residual, ϵ_r=1e-2.
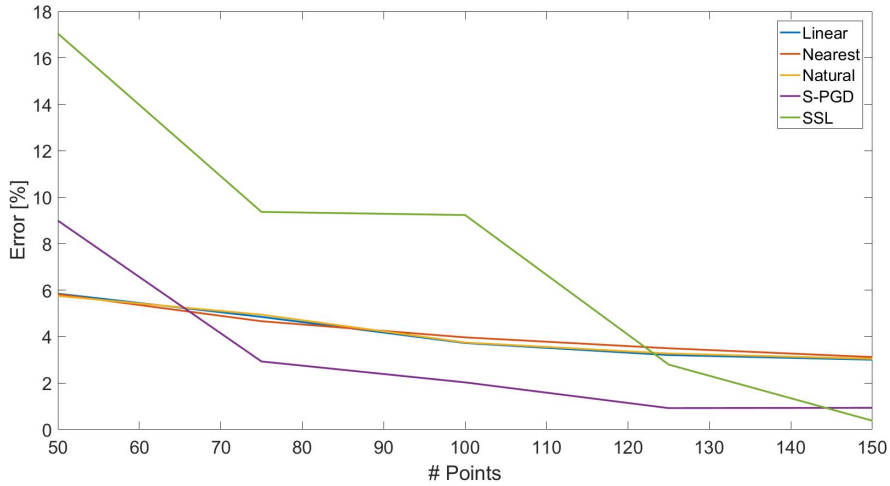
Figure 3.5 – $\mathcal{E}_{\mathcal{L}}$ of $f_1(x, y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

3.1.3   Numerical results

The aim of this part is to compare the ability of sparse model identification for different interpolation techniques. On one hand, the performance of standard techniques based on Delaunay triangulation such as linear, nearest neighbor or cubic interpolation is compared. Even though these techniques are simple, they allow to have a non-structured sampling point set since they rely on a Delaunay triangulation. On the other hand, the results are compared to the Sparse Subspace Learning (SSL) [Borzacchiello *et al.* 2017a]. The convergence and robustness of this method is proven to be very effective since the points are collocated at the Gauss-Lobato-Chebychev points. However, two main drawbacks appear considering this method. The first one is that there is a high concentration of points in the boundary of the domain, so that this quadrature is meant for functions that vary mainly along the boundary. Indeed, if the variation of the function appears in the middle of the domain, many sampling points will be required to converge to the exact function. The second one is that the sampling points have to be located at specific points in the domain. The s-PGD method using simple Kriging interpolants will be compared as well.

The numerical results are structured as follows: first two synthetic 2D functions are analyzed; secondly, two 2D response surfaces coming from a thermal problem and a Plastic Yield function are reconstructed; finally, a 10D synthetic function is reconstructed by means of the s-PGD algorithm.

3.1.3.1   2D synthetic functions

The first considered function is $f_1(x, y)$, as introduced in the previous section. Fig. 3.5 shows the reconstruction error ($\mathcal{E}_{\mathcal{L}}$) of $f_1(x, y)$ for different sampling points. As it can be noticed, the s-PGD algorithm performs well for a wide range of sampling points. Nevertheless, the SSL method is the one presenting the lower error level when there are more than 150 sampling points.
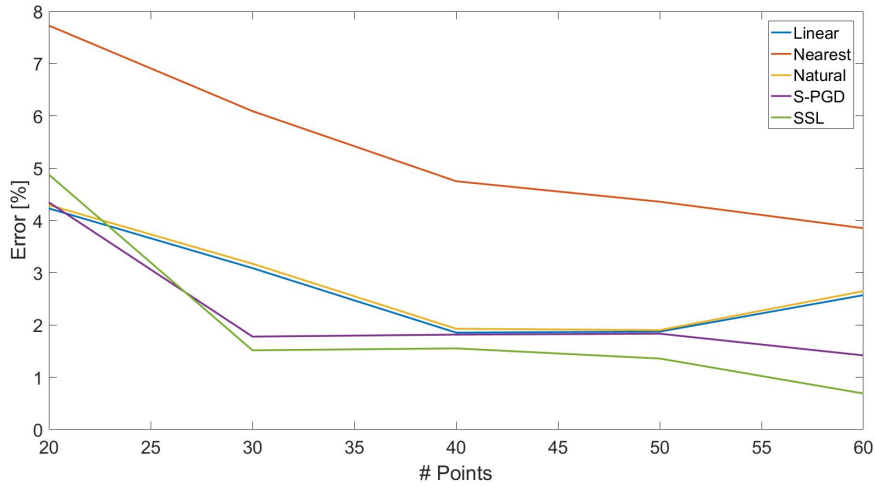
Figure 3.6 – $\mathcal{E}_{\mathcal{L}}$ of $f_2(x,y)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$.

A second synthetic function is defined as

$$f_2(x,y) = \cos(3xy) + \log(x + y + 2.05) + 5.$$

This function is intended to be reconstructed in the domain $\Omega_x = \Omega_y = [-1,1]$. It was chosen in such a way that it is relatively smooth in the center of the domain, whereas the main variation is located along the boundary of the domain. Indeed, this function is meant to show the potential of the SSL technique.

Fig. 3.6 shows the reconstruction error of the $f_2(x,y)$ function for different interpolation techniques. As it can be noticed, both SSL and s-PGD methods are the ones that present the best convergence properties. If the number of points is increased even more, the SSL method is the one that presents the lowest interpolation error. They are followed by linear and natural neighbor interpolations. Finally, the nearest neighbor method is the one presenting the worst error for this particular case.

3.1.3.2 2D response surfaces coming from physical problems

Once the convergence of the methods have been unveiled for synthetic functions, it is very interesting to analyze the power of the former methods by trying to identify functions that are coming from either simulations or models popular in the computational mechanics community. Indeed, two functions will be analyzed: the first one is an anisotropic Plastic Yield function, whereas the second one is a solution coming from a quasi-static thermal problem with varying source term and conductivity.

Fig. 3.7 shows the Yld2004-18p anisotropic plastic yield function, defined by Barlat et al. in [?]. Under plane stress hypothesis, this plastic yield function is a convex and closed surface defined in a three-dimensional space. Therefore, the position vector of an arbitrary point in the surface can be easily parameterized in cylindrical coordinates as $R(\theta, \sigma_{xy})$. The $R(\theta, \sigma_{xy})$ function for the Yld2004-18p is shown in Fig. 3.8, where anisotropies can be easily seen. Otherwise, the radius function will be constant for a given σ_{xy}.
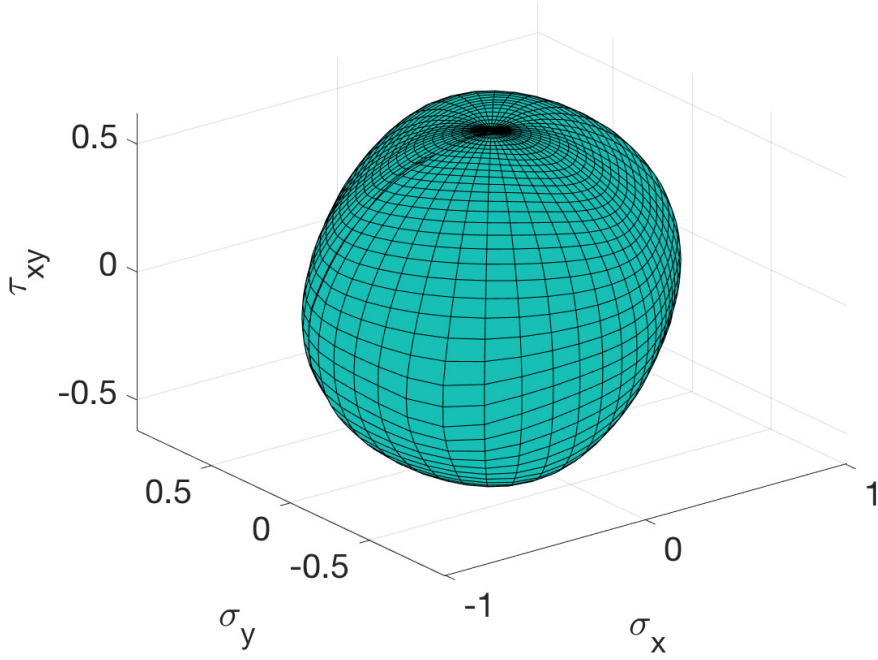
Figure 3.7 – Barlat's Yld2004-18p function under plane stress hypothesis.

Fig. 3.9 shows the error in the identification of the Barlat's plastic yield function Yld2004-18p. As it can be noticed, the s-PGD technique outperforms the rest of techniques. Indeed, the s-PGD is exploiting the fact that the response surface is highly separable.

As mentioned above, the second problem is the sparse identification of the solution of a quasi-static thermal problem modeled by

$$\nabla \cdot (\eta(x,t)\nabla(u(x,t))) = f(t), \qquad \text{in } \Omega_x \times \Omega_t = [-1,1] \times [-1,1], \tag{3.9}$$

where conductivity varies in space-time as

$$\eta(x,t) = (1 + 10\,\text{abs}(x) + 10x^2)\log(t+2.5) \qquad\qquad u(1,t) = 2 \tag{3.10}$$
$$f(x,t) = 10\cos(3\pi t) \qquad\qquad u(-1,t) = 2, \tag{3.11}$$

and the source term varies in time. Homogeneous Dirichlet boundary conditions are imposed at both spatial boundaries and no initial conditions are required due to quasi-stationarity assumptions.

Fig. 3.10 shows the evolution of the temperature field as a function of space time for the set of Eqs. (3.9)-(3.11). It can be noticed how the variation of the temperature throughout time is caused mainly due to the source term. However, conductivity modifies locally the curvature of the temperature along the spatial axis. Symmetry with respect the $x = 0$ axis is preserved due to the fact that the conductivity presents a symmetry along the same axis.

Fig. 3.11 shows the performance of each one of the techniques when trying to reconstruct the temperature field from certain sampling points. As can be noticed, the s-PGD in conjunction with Kriging interpolants is the one that presents the fastest convergence rate to the actual function, which is considered unknown. It is followed by linear and natural
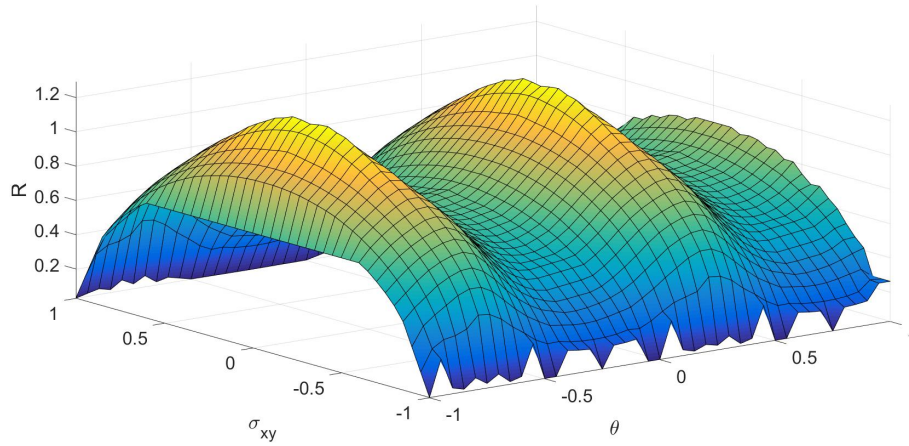
Figure 3.8 – $R(\theta, \sigma_{xy})$ function for Barlat's Yld2004-18p yield function.
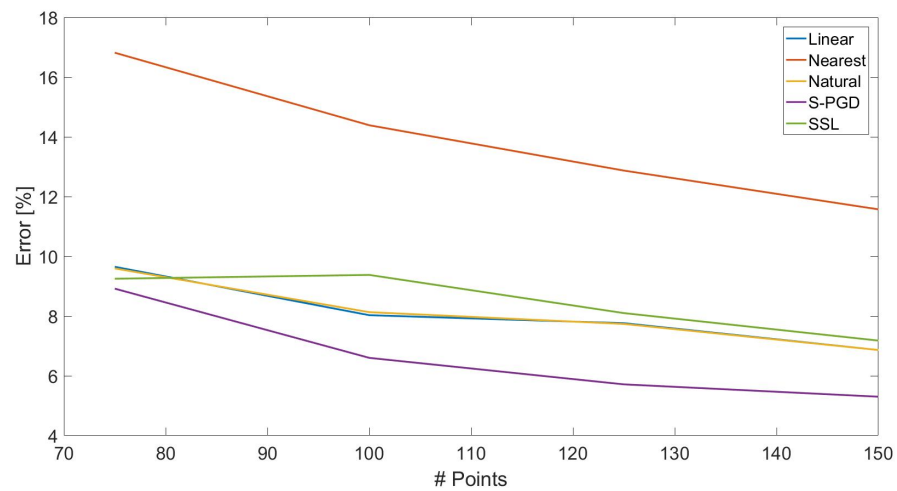
Figure 3.9 – $\mathcal{E}_{\mathcal{L}}$ of $R(\theta, \sigma_{xy})$ varying $\#\mathcal{P}$ for different sparse identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 5 \cdot 10^{-4}$.
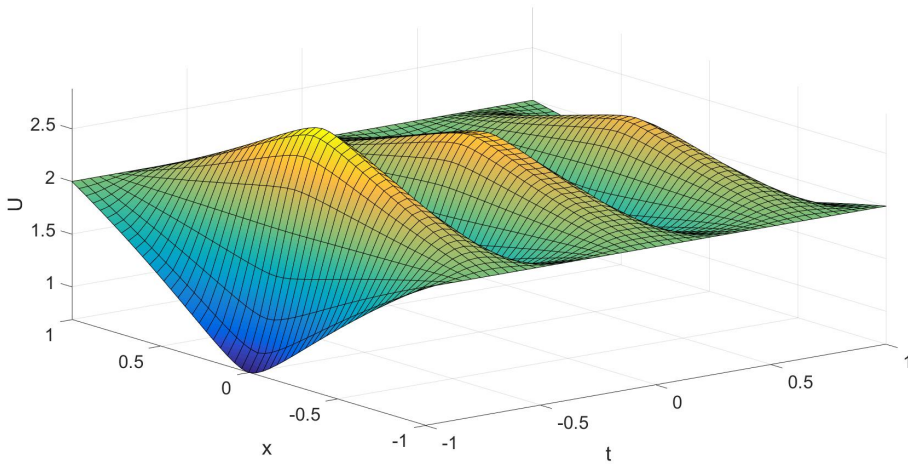
Figure 3.10 – Quasi-static solution to the thermal problem $u(x, t)$.

interpolations. The SSL method presents a slow convergence rate in this case, due to the fact that the main variation of the function $u(x, t)$ is happening in the center of the domain and not in the boundary.

3.1.3.3   A 10D multivariate case

In this part, we would like to show the scalability that s-PGD presents when dealing with relatively high-dimensional spaces. Since our solution is expressed in a separated format, an N dimensional problem (ND) is solved as a sequence of N 1D problems, which are solved using a fixed-point algorithm in order to circumvent the non-linearity of the separation of variables.

The objective function that we have used to analyze the properties of the s-PGD is defined as

$$f_3(x_1, x_2, \ldots, x_N) = 2 + \frac{1}{8} \sum_{i=1}^{N} x_i + \prod_{i=1}^{N} x_i + \prod_{i=1}^{N} x_i^2,$$

with $N = 10$ in this case.

Fig. 3.12 shows the error convergence in both sampling points ($\mathcal{E}_\mathcal{P}$, asterisks) and points out of the sampling ($\mathcal{E}_\mathcal{L}$, filled points). The $\mathcal{L}$ data set was composed by 3000 points, the $\mathcal{P}$ data subset for the s-PGD algorithm was composed by 500 points. The number of points required to properly capture the hyper-surface has increased with respect to the 2D examples due to the high dimensionality of the problem. Special attention has to be paid when increasing the cardinality of the interpolant basis without many sampling points, because the problem of high oscillations outside the control points may be accentuated.

3.1.4   Conclusions

In this work we have developed a data-based sparse reduced-order regression technique under the Proper Generalized Decomposition framework. This algorithm combines the robustness typical of the separation of variables together with properties of collocation
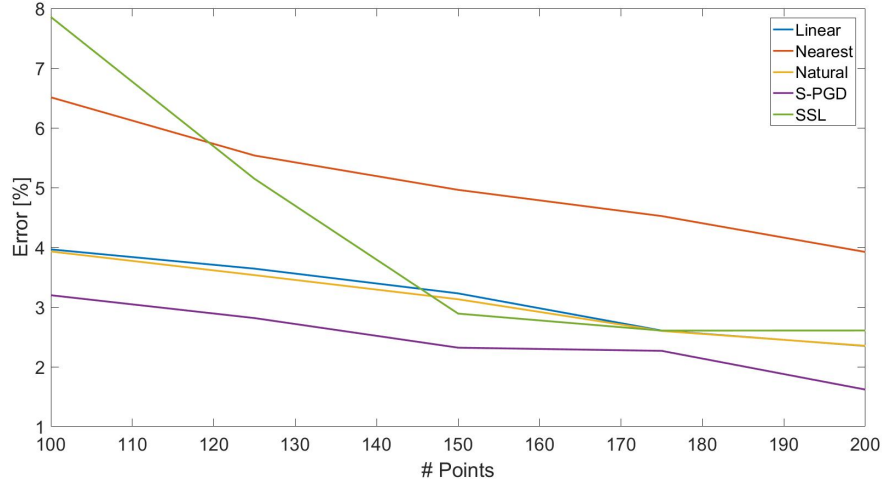
Figure 3.11 – $\mathcal{E}_\mathcal{L}$ of $u(x,t)$ varying $\#\mathcal{P}$ for different identification techniques. $\#\mathcal{L} = 1000$. $\epsilon_r = 2.5 \cdot 10^{-3}$.
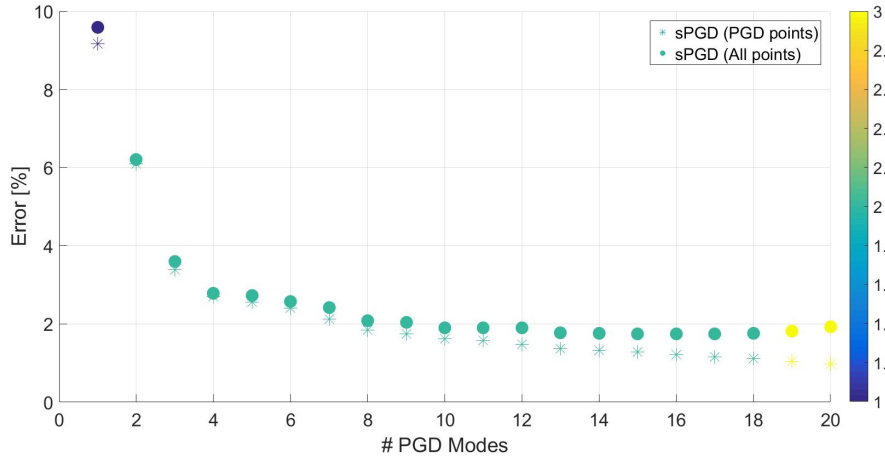
Figure 3.12 – $\mathcal{E}_\mathcal{L}$ (points) and $\mathcal{E}_\mathcal{P}$ (asterisk) versus the number of modes for $f_3(x_1, x_2, \ldots, x_N)$, $\#\mathcal{P} = 500$, $\#\mathcal{L} = 3000$. Modal adaptivity based on the residual, $\epsilon_r = 1e - 3$.

methods in order to provide with parsimonious models for the data at hand. The performance of simple Kriging interpolation has proven to be effective when the sought model presents some regularity. Furthermore, a modal adaptivity technique has been proposed in order to avoid high oscillations out of the sampling points, characteristic of high order interpolation methods when data is sparse.

The sPGD method has been compared advantageously versus other existing methods for different example functions. Finally, the convergence of sPGD method for a high dimensional function has been demonstrated as well.

3.2 sPGD Industrial Applications

This section aims to portrait three different applications of industrial interest in which the sPGD technique explained in section 3.1 has been succesfully applied. These three applications are listed below:

- Subsection 3.2.1. On the use of MOR techniques applied to crash simulations. The industrial partner Gestamp is interested in optimal car structural design since it plays an important role to guarantee safety requirement whenever a crash occurs. In this work, the sPGD methodology is applied to generate response surfaces of safety indicators which depend on several geometrical parameters. These response surfaces allow the industrial partner to modify the geometrical parameters to meet safety requirements while not exceeding certain mass constraints. The data was provided by Gestamp, generated with a high fidelity simulation using Pam-Crash solver owned by ESI Group.

- Subsection 3.2.2. Hybrid constitutive modeling: data-driven learning of corrections to plasticity models. The industrial partner ESI Group is interested in correcting existing plasticity models based on observable quantities of interest, providing physical error indicators. Therefore, the role of the sPGD in this scenario is to provide a surrogate model of the physical error indicator as a function of the parameterized plastic yield function correction. The best correction to the existent plastic model is found by locating the minimum of the just created surrogate model.

- Subsection 3.2.3. Prediction of composites behavior undergoing an automated tape placement (ATP) process through data-mining. In this case, as a part of a collaboration between Ecole Centrale de Nantes and Airbus group, there was a common interested into linking the geometrical parameters defining a given surface with the time needed to consolidate a new composite ply in a ATP process. The sPGD is used to generate a surrogate model of the consolidation time with respect the geometrical parameters defining the surface. The sPGD regression is compared with standard data-mining tools as it is the case of regression trees.

3.2.1 On the use of MOR techniques applied to crash simulations

When designing a new car prototype, it is important to ensure certain safety standards to avoid lethal injuries whenever a crash occurs. Since manufacturing a new car model is extremely expensive, simulation becomes a very powerful tool when optimizing the structure of the car. Fig. 3.13 portraits the intrusion towards the car symmetry plane when a lateral crash occurs. Indeed, minimizing the intrusion of the external car structure inside the interior cavity while limiting the maximum structure weight is of crucial interest.
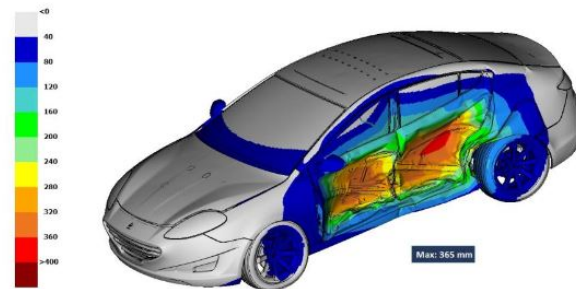
Figure 3.13 – Simulation of a lateral crash test. Contour intrusion towars the car symmetry plane.
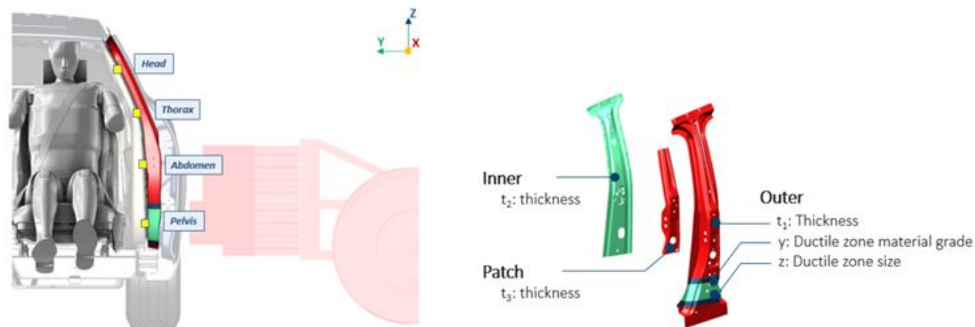

Figure 3.14 – Location of B-Pillar structure inside the car.

Fig. 3.14 shows two different views of a B-Pillar structure. This structure is placed between the two lateral doors and its main functionality is to reduce the intrusion inside the car whenever a crash occurs.

Unluckily, the cost of performing a high fidelity crash simulation is relatively expensive, thus, exploring a wide range of possible B-Pillar designs becomes unaffordable in practice. In this particular case, the influence of five different geometrical parameters (thickness and locations of B-Pillar subparts) is analyzed. Instead of following classical optimization procedures based on parameter sensitivities, a response surface of several Quantities of Interest (QoI) as a function of the five geometrical parameters is seeked.

Regarding the QoI, four different spatial points placed at different heights on this B-Pillar structure will be our object of study. This four points will be named in decreasing height order Upper (U), Middle Upper (MU), Middle Lower (ML) and Lower (L). Indeed, this four points are placed at the same height than the head, thorax, abdomen and pelvis of the passenger, respectively. Avoiding excessive intrusions of these four spatial points is compulsory to guarantee that the passengers will not suffer external injuries coming from the deformation of the vehicle structure. On the other hand, minimizing velocities/ accelerations also play a fundamental role to avoid internal injuries. Therefore, the capability to predict both intrusions and velocities of a new B-Pillar design whenever a crash occurs is of crucial interest.

Point	z	t_2	t_1	t_3	y
1	108	1.3	1.5	1.5	1.10
2	108	1.15	1.7	1.5	1.38
3	107	1.25	1.7	1.1	1.28
4	107	0.9	1.65	1.05	1.23
5	107	1.25	1.55	1.15	0.90
6	107	1.3	1.15	1.55	0.83
7	108	1.2	1.15	1.4	1.33
8	108	1.25	1.7	1	0.85
9	108	1.1	1.75	1.1	1.38
10	108	1	1.1	1.05	0.98
11	108	0.95	1.15	1.55	1.23
12	108	0.9	1.65	1.55	1.14
13	108	1.2	1.2	1.6	0.84
14	107	0.95	1.6	1.1	0.82
15	107	1.1	1.35	1.5	1.30
16	108	1.1	1.4	1.25	0.81
17	108	1.3	1.1	1.1	1.15
18	108	1.1	1.65	1.55	0.91
19	107	1.3	1.1	1.25	1.10
20	107	0.9	1.15	1.2	1.25
21	107	1	1.2	1.35	0.92
22	107	1.3	1.6	1.6	1.24

Table 3.1 – Set of Fixed Parameters for DoE

The parameter space consists of five independent geometrical parameters, namely z, t_2, t_1, t_3 and y. Fig. 3.14 (right) indicates the geometrical interpretation of each one of the parameters, t_1, t_2 and t_3 are related to the thicknesses of each substructure composing the B-Pillar, whereas z and y makes reference to the length of a ductile area located at the bottom part of the outer substructure. The Design of Experiments (DoE) consists on 22 high fidelity simulations, done with PAM-Crash solver from ESI Group, where different values of the independent parameters are set as shown in table 3.1.

Having said that, the first response surface involves only the maximal intrusion throughout the simulation time, thus, time is not a parameter. The second response surface concers the temporal evolution of the intrusion along the time coordinate, hence, the time is treated as an extra coordinate. Finally, the temporal velocity evolution at three points inside the B-Pillar is analyzed as well. These three points are placed at a height corresponding to the thorax, abdomen and pelvis of the passenger.

3.2.1.1 Maximum Intrusion

Fig. 3.15 shows the temporal evolution at the four positions of the B-Pillar structure for each one of the DoE appearing in Table 3.1. As it can be seen all of them follow the main
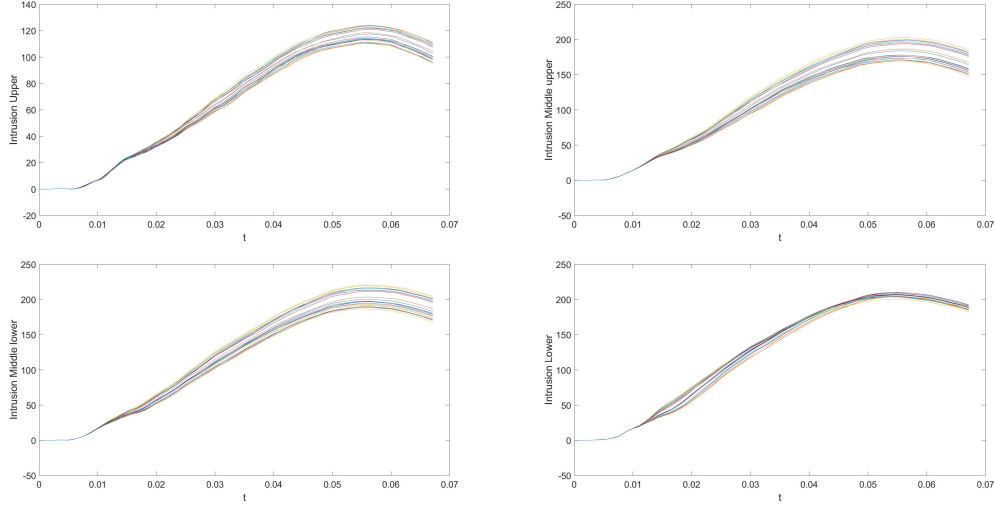
Figure 3.15 – Temporal evolution of the intrusion at four different locations of the B-Pillar structure.

trend, from 0 to 0.05 seconds there is a increment on the intrusion value due to the crash, afterwards there is a relaxation of the intrusion once the main impact has finished due to springback effects.

Since the maximum intursion of the B-Pillar is going to provide a reliable safety indicator, the first surrogate model is based on selecting the maximum intrusion at the U, MU, ML, L points as a function of the five parameters defined as shown in eq. 3.13. To shorten the notation, the whole set of parameters will be defined as shown in eq. 3.12.

$$\mathbf{p} = [z, \ t_2, \ t_1, \ t_3, \ y] \tag{3.12}$$

$$I_M^A(\mathbf{p}) = \max_{\forall t} I(t; \mathbf{p}, \mathbf{x}_A) \tag{3.13}$$

Where point A is valid for U, MU, ML, L points.

To test the performance of the sPGD algorithm, only the maximum intrusion related to points $[1, 4, 13, 15, 17, 21]$ inside the DoE are taken into account to construct the regression. The other 16 points are used as an error indicator to show how accurate the regression is. It is important to remark that the low amount of data to build the sPGD regression forces the algorithm to work with low order interpolation basis. Indeed, when generating such a surrogate model, up to linear interpolations in each one of the directions will be used. More in detail, the Greedy nature of the sPGD algorithm is used to adapt the basis for each one of the modes i.e. the first mode will be constant in each direction, the second mode linear in the first direction and constant in the other ones, the third mode linear in the second direction and constant in the other ones, etc.

Fig. 3.16 shows the real versus the estimated prediction on the maximum intrusion for Upper, Middle Upper, Middle Lower and Lower points i.e. I_M^U, I_M^{MU}, I_M^{ML}, I_M^L, respectively. The yellow points correspond to the ones used to build the sPGD regression, whereas the blue points are just used to measure the power of inference of the sPGD regression. If all points were on the red line, the surrogate model would be perfect. Nevertheless, the dispersion of these points with respect to the red line gives us a visual indicator of how
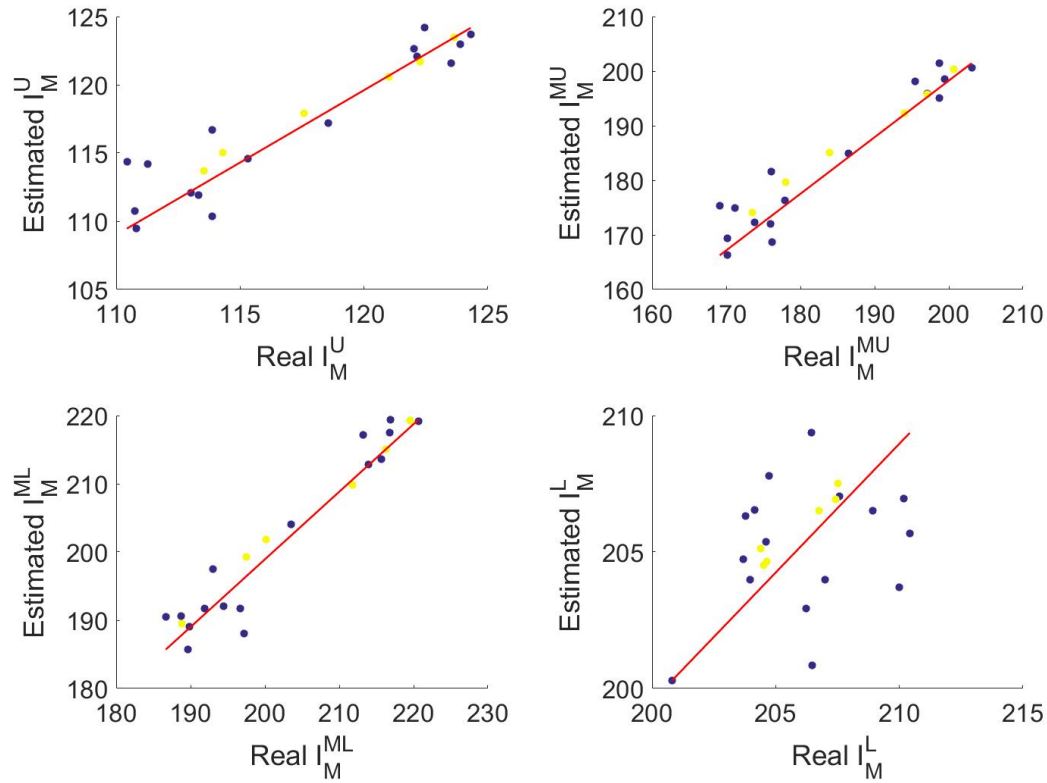
Figure 3.16 – Estimated versus real maximum intrusion for Upper (U), Middle Upper (MU), Middle Lower (ML) and Lower (L) B-Pillar points. Yellow points, used in the sPGD. Blue points, used as error indicator of the regression model.

good the surrogate model is. Indeed, the relative error based on the blue points is shown in Table 3.2. As it can be seen, the highest error is the one present in the Lower point, reaching a value of 4.8%.

3.2.1.2 Temporal Evolution of the Intrusion

The previous section analyzed the surrogate models based on the maximum intrusion. However, it is also important to understand how the intrusion involves in time. Indeed, this temporal evolution gathers the information of the maximum intrusion just like the cumulated energy stored in the B-Pillar structure, which may be interesting to predict possible failures of the structure. Nevertheless, the generation of the surrogate model involving the time coordinate becomes more complex.

-	Upper	Middle Upper	Middle Lower	Lower
Rel. Error [%]	1.35	1.73	1.38	4.87

Table 3.2 – Relative Error in maximum intrusion for s-PGD when considering 6 DoE out of 22.

Hence, the four QoI treated in this subsection reads as shown in eq. 3.14, where the point $\mathbf{x}_A$ stands for either Upper, Middle Upper, Middle Lower or Lower. To be consistent with the former subsection, in order to build the sPGD model only the time snapshoots coming from points $[1, 4, 13, 15, 17, 21]$ are taken into consideration. It is important to reckon that the time coordinate experiments a rather complex behaviour compared to the evolution along the other coordinates. However, due to the fact that more points are placed along the time coordinate (i.e. the entire time snapshoot of the i-th DOE point is taken), it is possible to increase the interpolation order along this time coordinate while keeping low order in the other coordinates. For this reason, Chebychev polynomials of degree 40 are used in the time direction, while keeping either constant or linear interpolation in the $\mathbf{p}$ space.

$$I^A(\mathbf{p}, t; \mathbf{x}_A) \tag{3.14}$$

Fig. (3.17) portraits the both the error considering only the points inside the sPGD regression and the one considering both sPGD regression points plus the ones outside the trainning dataset. As it can be seen, the error decreases as a function of the number of modes introduced in our sPGD approximation. The most important decay in the error is seen in the fourth sPGD which involves a linear interpolation along p_3, hence it can be assumed that this parameter plays an important role in order to explain the variation of the QoI inside the parameter space.

Fig. (3.18) shows the predicted temporal evolution (red) against the real temporal evolution (blue) for a point inside the trainning set (i.e. DOE 1 left) and outside the trainning set (i.e. DOE 22 right). As it can be seen, there is almost no difference between predicted and real values for the point in the data set, i.e. blue and red curves overlap, whereas for the point outside the data set there is a very slight difference between red and blue curves. Nevertheless, points outside the trainning set present acceptable errors in this particular case.

Analogously, figs. 3.19-3.21 contains the same information than Fig. 3.18 but in the case of Middle Upper, Middle Lower and Lower points, respectively. The predicted behaviour is in good agreement with respect to the real behaviour.

3.2.1.3 Temporal Evolution of the Velocity

Another quantity of interest that is important from a safety point of view is the velocity magnitude suffered at certain points belonging to the body of the passenger throughout the impact. Indeed, the velocity magnitude at three points placed at the thorax, abdomen and pelvis heights are our subject of study, namely $V_T(\mathbf{p}, t)$, $V_A(\mathbf{p}, t)$ and $V_P(\mathbf{p}, t)$, respectively.

Fig. (3.22) depicts the temporal evolution of the velocity magnitude throughout the crash for different set of parameters $\mathbf{p}$. As it can be seen, there is a notable change of this curves within the parameter space. Initially, the velocity magnitude is zero since the car is at rest, then it starts to grow due to the impact. Hence, the main task is to be able to predict such temporal evolution for different values of the parameter space, higher interpolation degree time against the other parameter directions is used since the variation along the time direction is expected to be harder to capture.

Fig. (3.23) shows the convergence of the estimated solution for different number of sPGD modes. The error is measured using a $L2$ relative error norm. The points used in the sPGD regression are $[1, 4, 13, 15, 17, 21]$, the other points are left to measure the performance of the regression. As it can be seen, the magnitude of all error indicators are acceptable (around 1%-2%) being the highest error at the pelvis velocity. Indeed, the pelvis velocity
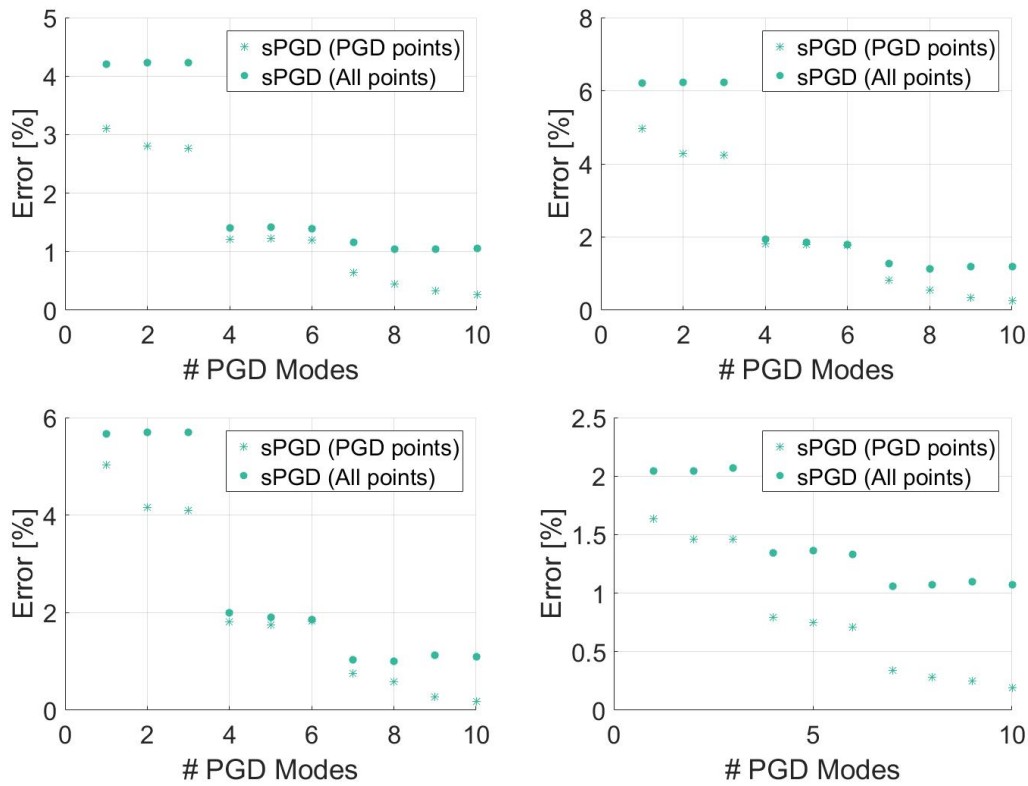
Figure 3.17 – Convergence with respect number of sPGD modes for Upper (Top-Left), Middle Upper (Top-Right), Middle Lower (Bottom-Left) and Lower (Bottom-Right)
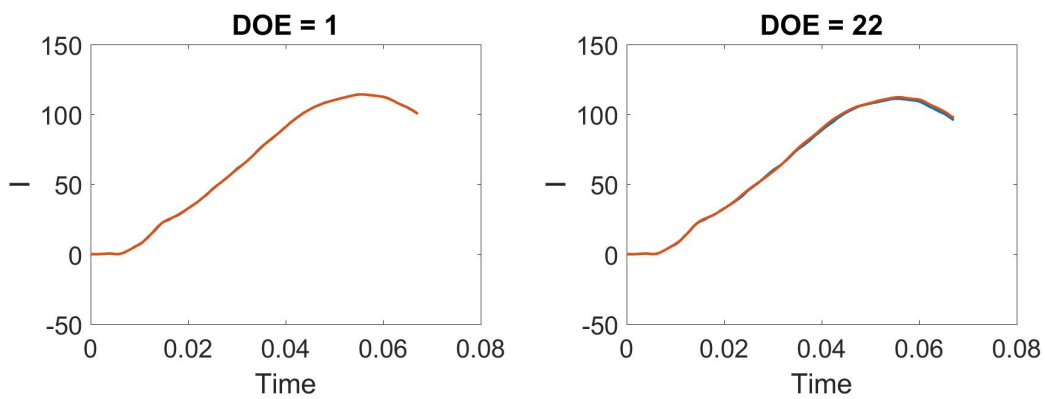

Figure 3.18 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a Upper point inside the trainning set (left) and outside the trainning set (right).
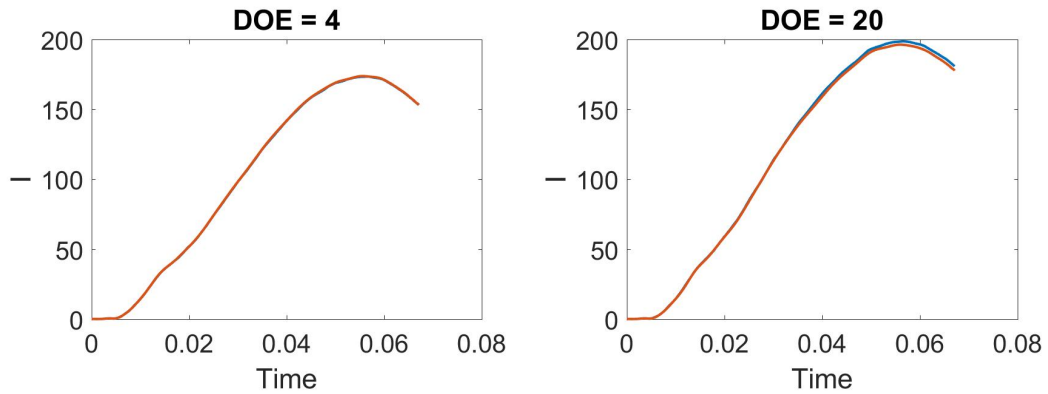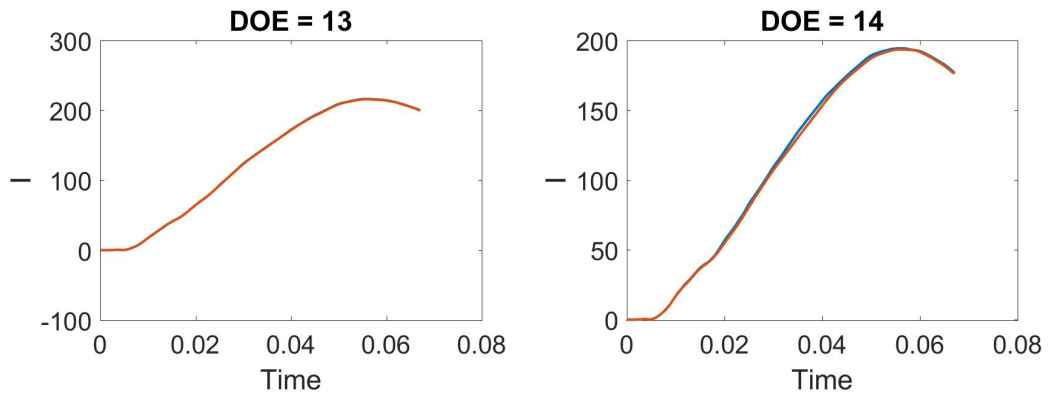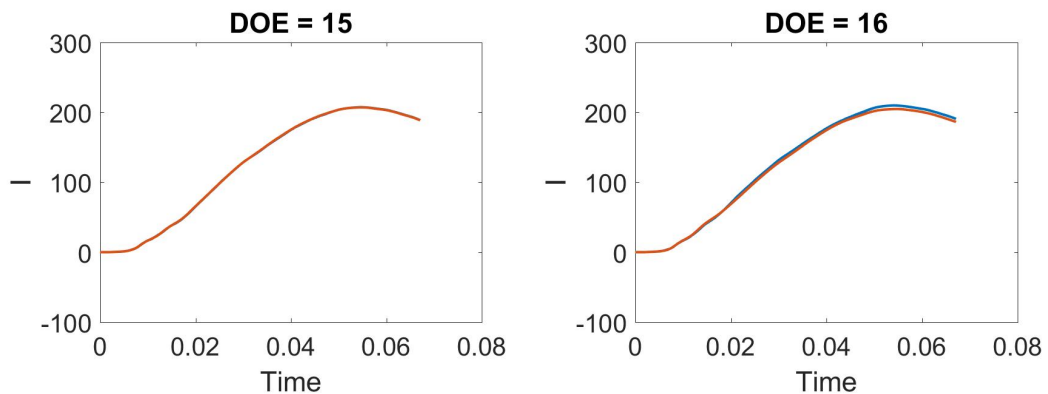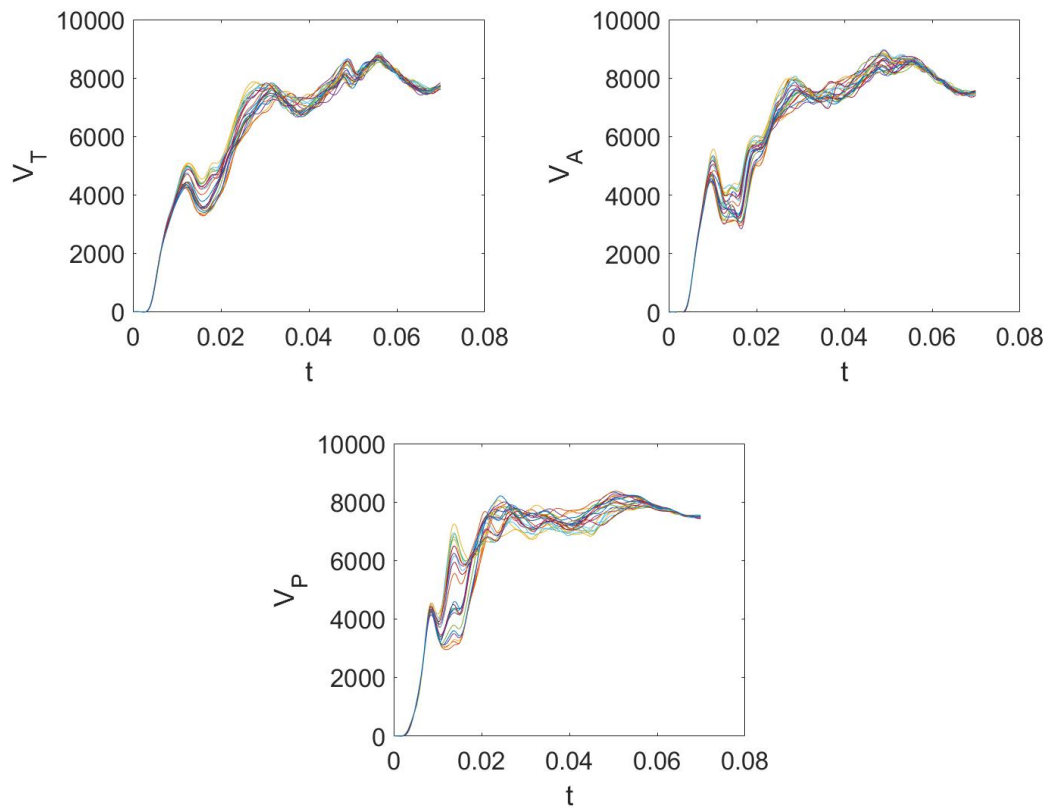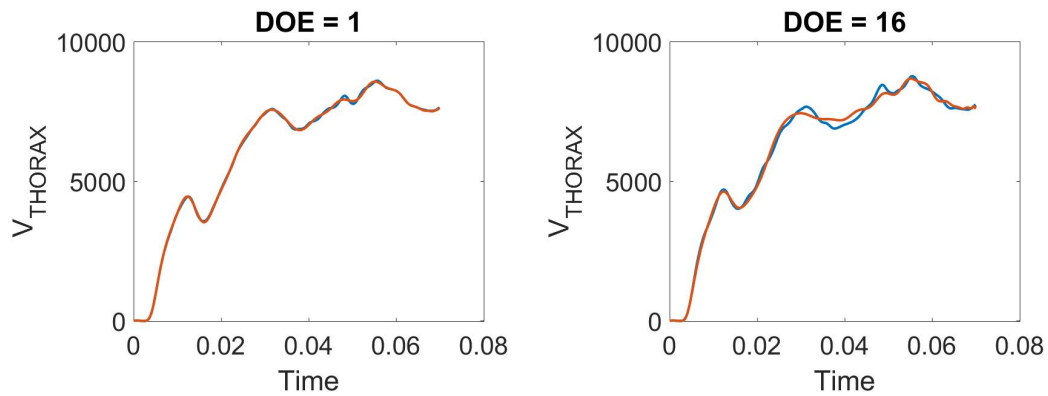
Figure 3.19 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a Middle Upper point inside the trainning set (left) and outside the trainning set (right).

Figure 3.20 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a Middle Lower point inside the trainning set (left) and outside the trainning set (right).
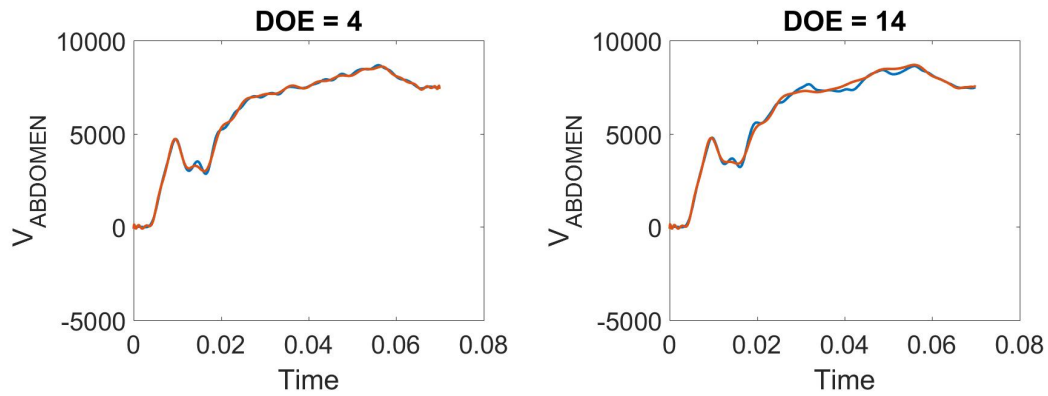

Figure 3.21 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a Lower point inside the trainning set (left) and outside the trainning set (right).

Figure 3.22 – Temporal evolution of the thorax, abdomen and pelvis velocity magnitude throughtout the crash simulation for different values of the parametric space.
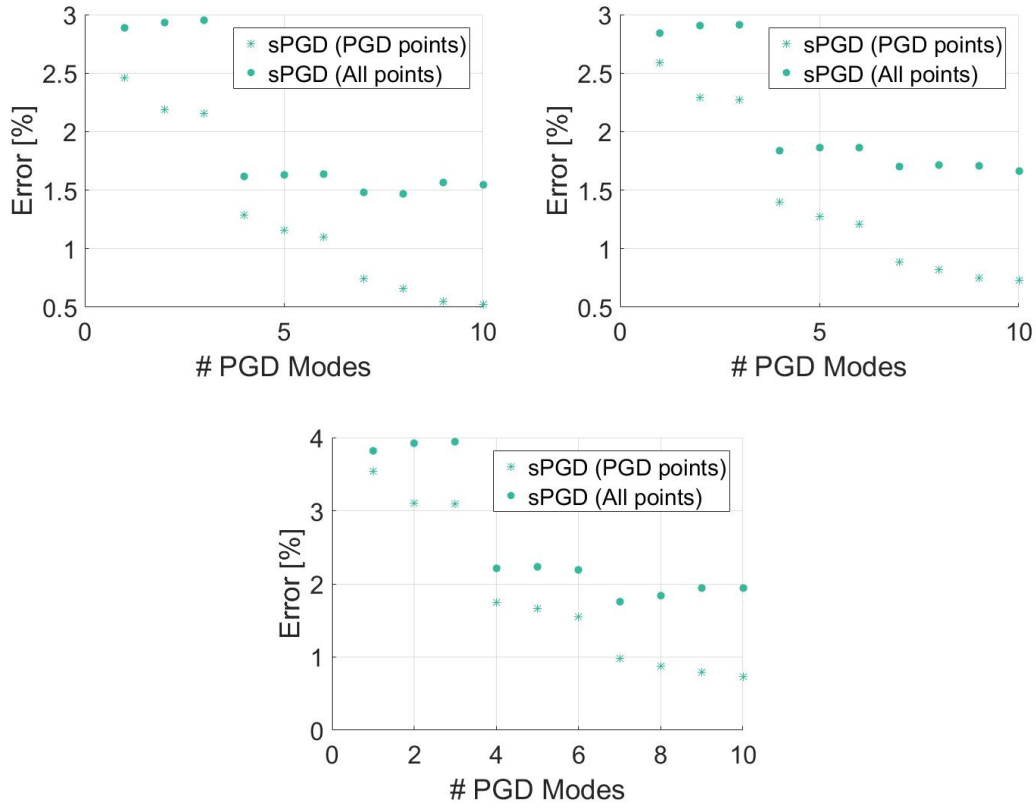
Figure 3.23 – Convergence with respect number of sPGD modes for thorax velocity (Top-Left), abdomen velocity (Top-Right) and pelvis velocity (Bottom)

is the one presenting the highest error since the variation of the curve along the parameter space is the highest one as well.

Figs. (3.24)-(3.26) shows the predicted temporal evolution (red) versus the real temporal evolution (blue) for velocity magnitudes inside the trainning set (left) and outside the trainning set (right). It can be notice how the predicted curve is closer to the real curve when the point is inside the trainning data set. Nevertheless, the prediction outside the trainning point also captures the main characteristics of the real curve.

3.2.1.4 Conclusions

In the light of the results, the sPGD methodology could be seen as an alternative route to create a response surface related to crash simulations. These response surfaces are ultimately used by the engineers in order to decide whenever a variation of a given parameter is important to satisfy the problem constraints. In this particular case, 22 high fidelity simulations are performed, whereas only 6 high fidelity simulations are used to construct the sPGD regression model. Considering the low error committed at the DoE points left outside the regression model confirms that only few high-fidelity simulations are required to build a reasonable response surface.

Figure 3.24 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a thorax velocity magnitude inside the trainning set (left) and outside the trainning set (right).

Figure 3.25 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a abdomen velocity magnitude inside the trainning set (left) and outside the trainning set (right).
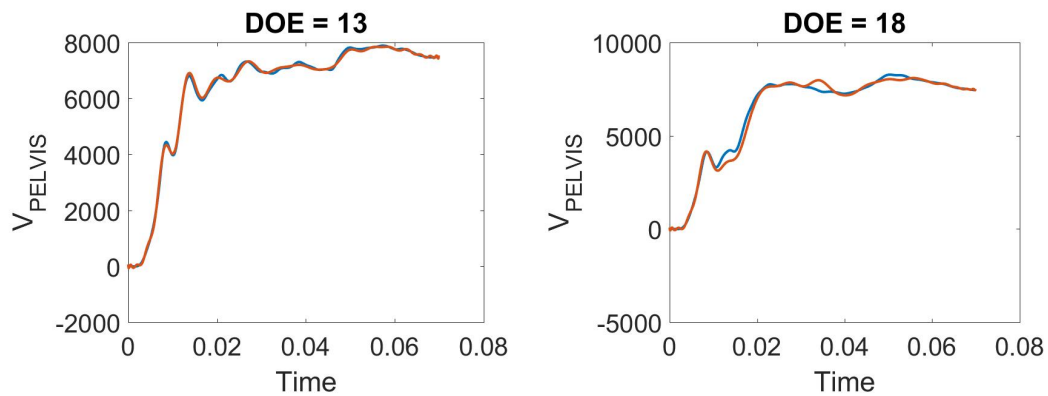
Figure 3.26 – Predicted temporal evolution (red) versus real temporal evolution (blue) for a pelvis velocity magnitude inside the trainning set (left) and outside the trainning set (right).

3.2.2 Hybrid constitutive modeling: data-driven learning of corrections to plasticity models

Plenty of effort has been dedicated throughout history to create very accurate models, as an example the reader may think about all different models formulated, for instance, in hyperelasticity like Neo-Hookean, Ogden, Saint-Venant. Another framework in which there are plenty of constitutive models is the one related to plasticity where we can highlight Tresca, Von Misses or Hill criterions among others. However, we also know that no model is perfect, it is always subjected to certain hypothesis. Indeed, even if you can calibrate a model perfectly well, no garantee is given that for another set of experiences different from the calibration ones, the model is going to provide you a perfect result.

The main aim of this work is to provide an alternative route to enhance existing models including information coming from data, the so called data-driven correction. Particularly, special effort is placed into the calibration of plastic yield functions.

The data driven correction is quite simple, imagine that our departure point is a given model $\mathcal{M}(\mathbf{p})$. It is important to keep in mind that we are looking for a enhancement of the previous model, therefore a divergence model $\mathcal{D}(\mathbf{c})$, which applies to the first model, has to be defined. Therefore, reality ($\mathcal{R}$) is tried to be approximated as shown in eq. (3.15).

$$\mathcal{R} \approx \mathcal{M}(\mathbf{p}) + \mathcal{D}(\mathbf{c})\big|_{\mathbf{p}} \tag{3.15}$$

If we want to be truly objective, since our measurement capabilities will be constraint to some observable quantities, everything will depend from the set of experiences or observable quantities $\mathcal{S}$, as shown in eq. (3.16).

$$\mathcal{R}\big|_{\mathcal{S}} \approx \mathcal{M}(\mathbf{p}) + \mathcal{D}(\mathbf{c})\big|_{\mathbf{p},\mathcal{S}} \tag{3.16}$$

It is worth to mention also that the way we define the observables $\mathcal{S}$ could have an important impact over the calibration of the set of parameters, $\mathbf{c}$. Ideally, a set of experiences such that the entire parametric space $\mathbf{c}$ is not singular has to be defined. Otherwise, it may lead to spurious modes in such parametric space.

3.2.2.1 Problem Statement

In the present work, we will try to capture the plastic yield function behind a given experiment. Just as a recall, a plastic yield function is a hypersurface living in the stress ($\boldsymbol{\sigma}$) space. Typically , this surface is parameterized using a finite set of parameters ($\mathbf{p}$) given by the first order model ($\mathcal{M}(\mathbf{p})$). Moreover, it will depend also on the correction model ($\mathcal{D}(\mathbf{c})$), therefore, a general plastic yield function can be written as shown in eq. (3.17).

$$\mathcal{F}_Y(\boldsymbol{\sigma}; \mathbf{p}, \mathbf{c}) = 0 \tag{3.17}$$

For the sake of simplicity, but without loosing generality, we will constraint our set of experiences to plane stress hypothesis. Therefore, our plastic yield function is defined in a three dimensional space corresponding to the three active stress components ($\sigma_{xx}, \sigma_{yy}, \tau_{xy}$). Moreover, it is relatively easy to express it in spherical coordinates as shown in eqs. (3.18), since the plastic yield function is normally a convex closed surface.

$$\begin{aligned}
\sigma_{xx} &= R cos(\alpha) sin(\beta) \\
\sigma_{yy} &= R sin(\alpha) sin(\beta) \\
\tau_{yy} &= R cos(\beta)
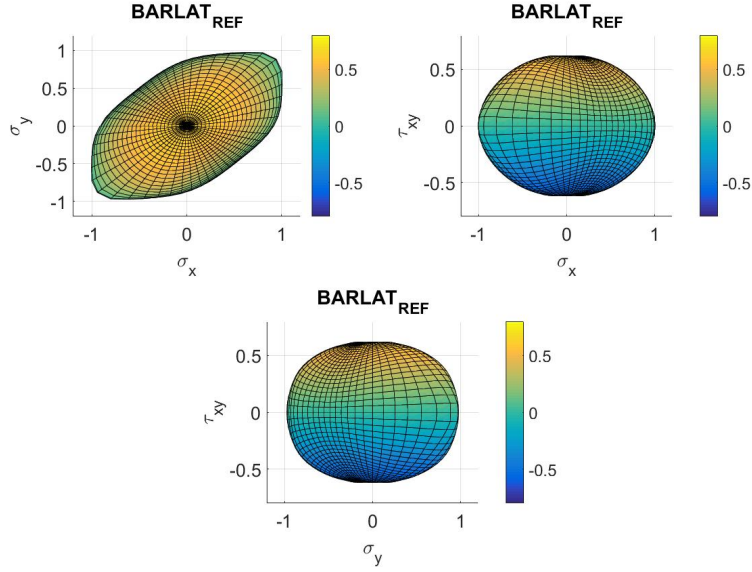\end{aligned} \tag{3.18}$$

Figure 3.27 – Different views of Barlat Yld2004-18p plastic yield function. Colours are τ_{xy}.

Where $R(\alpha, \beta)$ function defines the radius in spherical coordinates for any possible angle. Therefore, a parameterization of $R(\alpha, \beta; \mathbf{p})$ directly determines the shape of $\mathcal{F}_Y$ whenever a model without corrections is considered.

In this very first work, we will work with synthetic data, since the main objective is to show the potential of the algorithm. Thus, reality ($\mathcal{R}$) is modeled using a Barlat Yld2004-18p plastic yield function as shown in fig. (3.27).

The first model ($\mathcal{M}$) involves a quadratic Hill plastic yield function as shown in eq. (3.28). As it can be seen, this yield criterion presents a parameterization based on four coefficients, i.e. ($\#\mathbf{p} = 4$).

$$\mathcal{F}_Y^H(\sigma_{xx}, \sigma_{yy}, \tau_{xy}; F, G, H, N) = F\sigma_{yy}^2 + G\sigma_{xx}^2 + H(\sigma_{xx} - \sigma_{yy})^2 + N\tau_{xy}^2 - \sigma_0^2 \qquad (3.19)$$

Fig. (3.28) depicts the shape of a quadratic Hill yield criterion when $F = 2.1$, $G = 1.8$, $H = 0.7$ $N = 1.9$. As it can be seen, convexity is fulfilled and it defines a smooth closed surface in the stress space.

The divergence model ($\mathcal{D}(\mathbf{c})$) involves a correction of the first model. In this particular case, this correction is done involving a set of 8 control points distributed along the plane $\tau_{xy} = 0$, plus another degree of freedom which relates to the movement of the maximum shear points defined in the line ($\sigma_{xx} = 0$, $\sigma_{yy} = 0$), hence ($\#\mathbf{c} = 9$). The connexion between all degrees of freedom is done by means of a C1 continuous shape functions, also known as natural interpolation. Fig. (3.29) portraits the impact of moving one degree of freedom in $\mathcal{D}(\mathbf{c})$ on the quadratic Hill yield function depicted in fig. (3.28). As it can be seen, the maximum of the perturbation is achieved where the degree of freedom is placed and the magnitude is smoothly decreasing being minimum when the neighbouring degrees of freedom are reached.

Eq. 3.20 provides an error indicator which is able to quantify the discrepancies between the results simulating the set of experiences $\mathcal{S}$ and the results provided by the real exper-
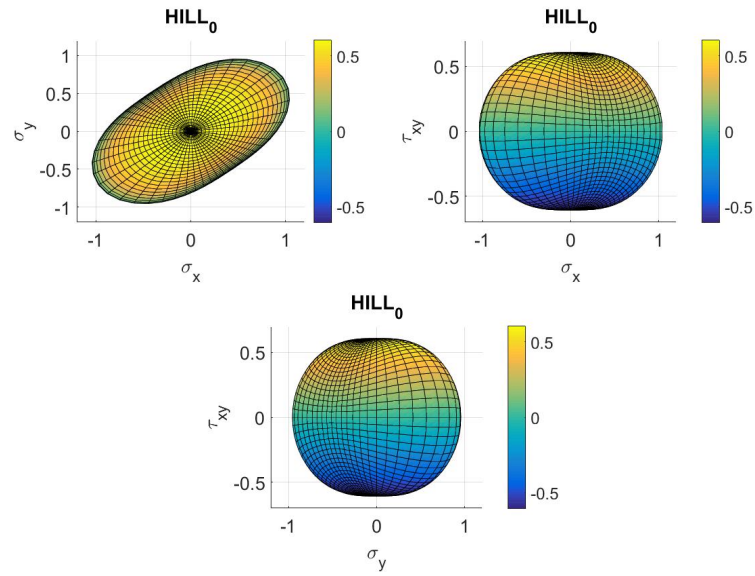
Figure 3.28 – Different views of quadratic Hill plastic yield function. Colours are τ_{xy}. $F = 2.1$, $G = 1.8$, $H = 0.7$ $N = 1.9$.
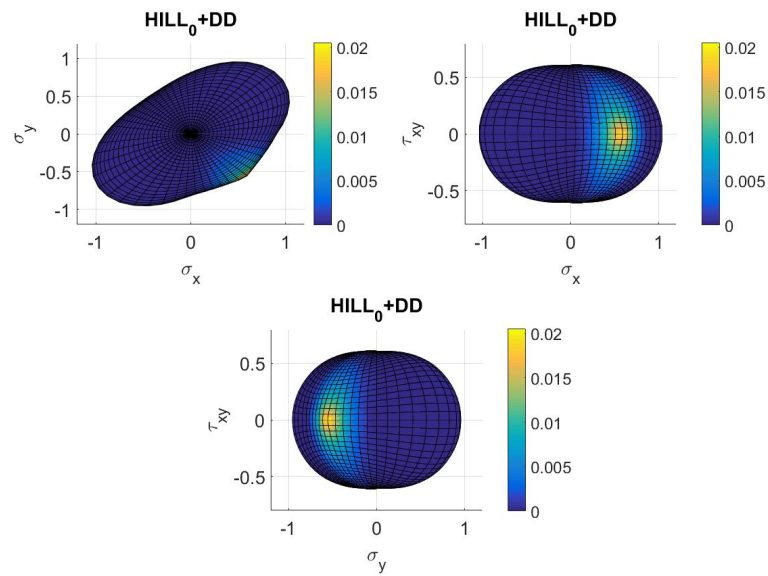

Figure 3.29 – Perturbed quadratic Hill plastic yield function moving only one dof of $\mathcal{D}(\mathbf{c})$. Colours are the magnitude of the perturbation. $F = 2.1$, $G = 1.8$, $H = 0.7$ $N = 1.9$.

iment. In this particular case, we select to measure the error using the strain field, but we could have done with the displacement field as well. If the first order model is already calibrated, the only parametric space that could vary is the one related to the divergence model.

$$E_{\mathcal{S}}(\mathbf{c}) = \sum_{s=1}^{\#\mathcal{S}} \int_{t_s} \int_{\mathbf{x}_s} ||\epsilon_{sim}^s(\mathbf{x}, t, \mathbf{c}) - \epsilon_{ref}^s(\mathbf{x}, t)|| d\mathbf{x} dt \qquad (3.20)$$

The main idea is to build a response surfaces ($E_{\mathcal{S}}(\mathbf{c})$) to characterize the parametric spaces based on a existing set of experiences. Once the response surface is built, the global minimum of the response surface is going to provide the best candidate in the parametric space which is able to reproduce the set of experiences. Moreover, it is important to recall that our reality ($\mathcal{R}$) comes from a simulation using a Barlat Yld2004-18p, therefore, even if this quantity is not used in practice, we can always measure the error in the reconstructed plastic yield surface as shown in eq. (3.21). Indeed, this last error indicator is used to check if our identified correction presents less deviation than the first order model.

$$E_{\mathcal{F}_Y}(\mathbf{c}) = \int_\alpha \int_\beta ||R_{ref}(\alpha, \beta) - R_{sim}(\alpha, \beta; \mathbf{c})|| d\alpha d\beta \qquad (3.21)$$

Indeed, a perfect set of experiences will be the one providing a monotonic ratio between $E_{\mathcal{F}_Y}(\mathbf{c})$ and $E_{\mathcal{S}}(\mathbf{c})$, ensuring that the minimum in the strain error measure corresponds with the minimum in the plastic yield error measure. Otherwise, it will mean that our set of experiences may not be sensitive to certain variations in the parametric space.

3.2.2.2 Numerical Results

Two different set of quadratic Hill criterions have been used as a starting point in our perturbation model, $\mathcal{H}_1$ and $\mathcal{H}_2$. The first one is already quite close to the Barlat's criterion, since it has a error $E_{\mathcal{F}_Y}^{\mathcal{H}_1} = 1.57$, whereas the second one presents an error of $E_{\mathcal{F}_Y}^{\mathcal{H}_2} = 24.9$. The parameter space to create both functions are $F_1 = 2.1$, $G_1 = 1.8$, $H_1 = 0.7$ $N_1 = 1.9$ and $F_2 = 2.3$, $G_2 = 2.0$, $H_2 = 0.8$ $N_2 = 1.7$, respectively. Fig. 3.30 portraits the error between the $\mathcal{H}_1$ criterion and the Barlat Yld2004-18p projected in the $\mathcal{H}_1$ criterion plastic surface.

Regarding the set of experiences ($\mathcal{S}$) to calibrate the models, we have chosen to use a set of relatively simple tests in a squared domain, $\Omega = [0,1] \times [0,1]$. The set of boundary conditions is specified in eqs. 3.22. Hence, varying the both tractions $\mathbf{t}_1$ and $\mathbf{t}_2$, different regions of the stress space inside the piece of solid are explored. Indeed, 40 different experiments have been included in order to create $E_{\mathcal{S}}(\mathbf{c})$.

$$\begin{aligned} u(0, y) &= 0 \\ v(x, 0) &= 0 \\ \mathbf{t}(1, y) &= \mathbf{t}_1 \\ \mathbf{t}(x, 1) &= \mathbf{t}_2 \end{aligned} \qquad (3.22)$$

In order to build the response surfaces $E_{\mathcal{S}}^{\mathcal{H}_1}(\mathbf{c})$ and $E_{\mathcal{S}}^{\mathcal{H}_2}(\mathbf{c})$, 1000 simulations, randomly selecting the parametric space, have been done for each case. Each realization of the parametric space follows a uniform distribution from $[-0.1, 0.1]$ in the $\mathcal{H}_1$ case and $[-0.15, 0.15]$ in the $\mathcal{H}_2$ case since we expect a major correction.
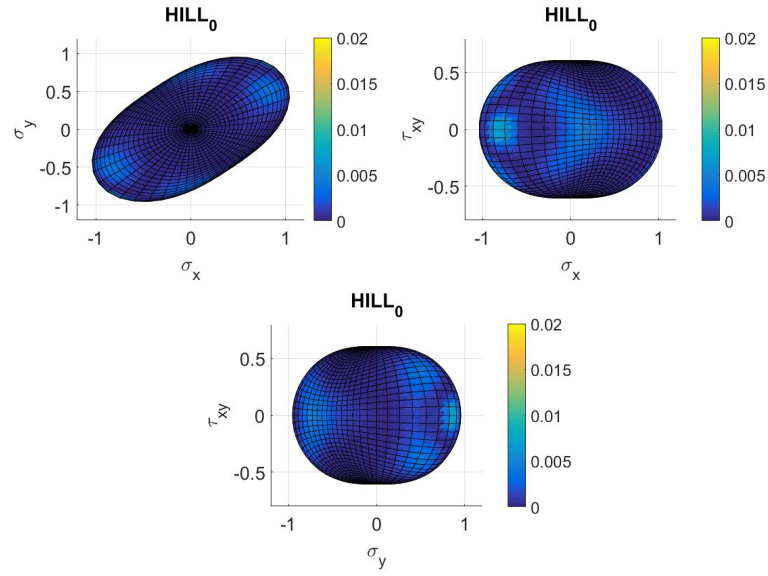
Figure 3.30 – Different views of $\mathcal{H}_1$ plastic yield function. Colours are missmatch between the $\mathcal{H}_1$ criterion and the Barlat Yld2004-18p. $E_{\mathcal{F}_Y}^{\mathcal{H}_1} = 1.57$.
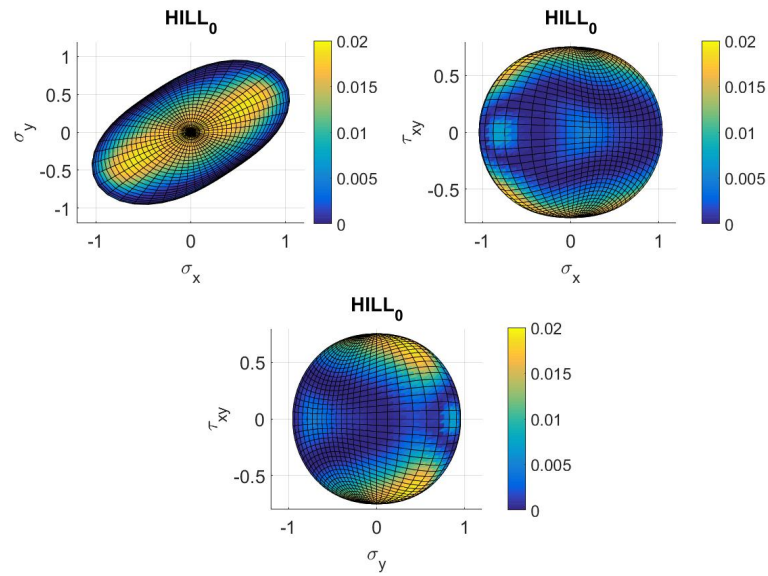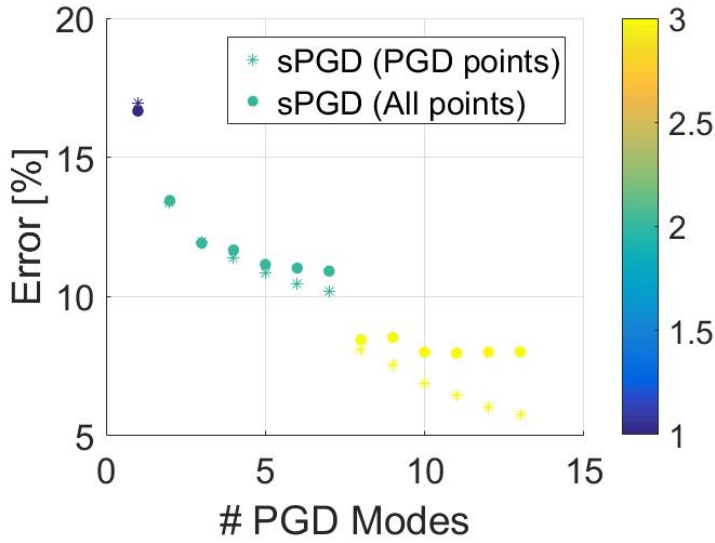

Figure 3.31 – Different views of $\mathcal{H}_2$ plastic yield function. Colours are missmatch between the $\mathcal{H}_2$ criterion and the Barlat Yld2004-18p. $E_{\mathcal{F}_Y}^{\mathcal{H}_2} = 24.9$.

Figure 3.32 – Error in the reconstruction $E_{\mathcal{S}}^{\mathcal{H}_2+DD}(\mathbf{c})$ using half of the points in the data base for trainning.

Several options could be adopted to reconstruct the entire response surface. Though non-structured interpolation techniques based on Delaunay triangularization can be used, they will suffer when the dimensionality of the parametric space increases. In this particular case, a non-linear regression technique called Sparse Proper Generalized Decomposition is used. It strongly relies on the separation of variables to circumvent the problem of high dimensional spaces, just like in a collocation method to build relatively smooth solutions from the available dataset. Indeed the main objective is to try to capture the whole response surface using as less points as possible. By doing that, instead of performing an standard minimization procedure, we will infer the minimum once the sPGD response surface is built. Therefore, the sPGD algorithm is applied to half of the points previously precomputed in $E_{\mathcal{S}}^{\mathcal{H}_2+DD}(\mathbf{c})$. The other half of the points are used to quantify the error in the prediction of the surface.

As it can be seen in fig. (3.32) the response surface presents 7 percent of mean relative error. This error could be decreased easily if more sample points are added to the sPGD algorithm.

Once the response surface has a continuous and separated representation, the minimum is searched by employing a line search sequentially in each one of the separated directions. Several initialization points at which the line search algorithm is started are selected randomly to almost ensure the globality of the minimum.

Fig. (3.33) shows the error when the yield surface $\mathcal{H}_1$ is corrected with the data driven correction. As it can be seen, the final reconstructed error $E_{\mathcal{F}_Y}^{\mathcal{H}_1+DD}$, has been reduced with respect to the $E_{\mathcal{F}_Y}^{\mathcal{H}_1}$ error, passing from 1.57 to 1.27. Indeed, a 19 per cent of improvement has been achieved with the data driven correction for this particular case.

Fig. (3.34) shows the error when the yield surface $\mathcal{H}_2$ is corrected with the data driven correction. In this particular case, the final reconstructed error $E_{\mathcal{F}_Y}^{\mathcal{H}_2+DD}$ has been reduced as well from 24.9 to 4.63. Therefore, a 81 per cent of improvement has been produced in this particular case.
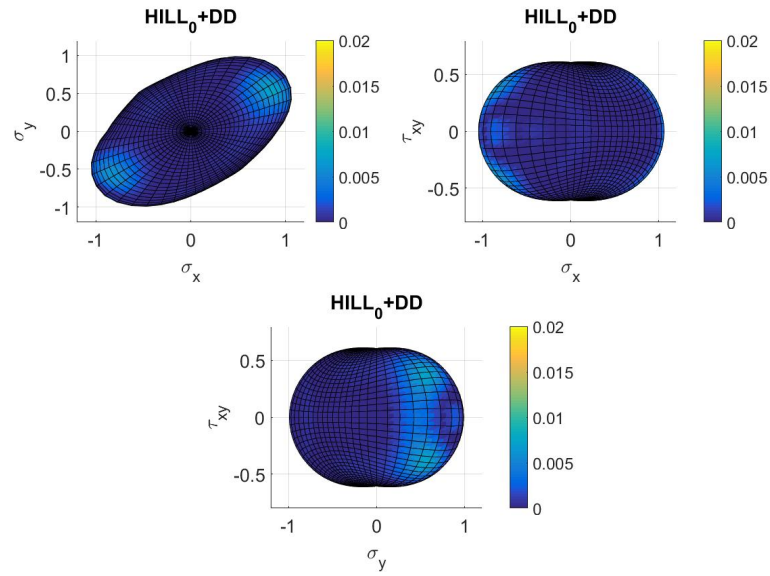
Figure 3.33 – Different views of $\mathcal{H}_1 + DD$ plastic yield function. Colours are mismatch between the $\mathcal{H}_1 + DD$ criterion and the Barlat Yld2004-18p. $E_{\mathcal{F}_Y}^{\mathcal{H}_1 + DD} = 1.27$.
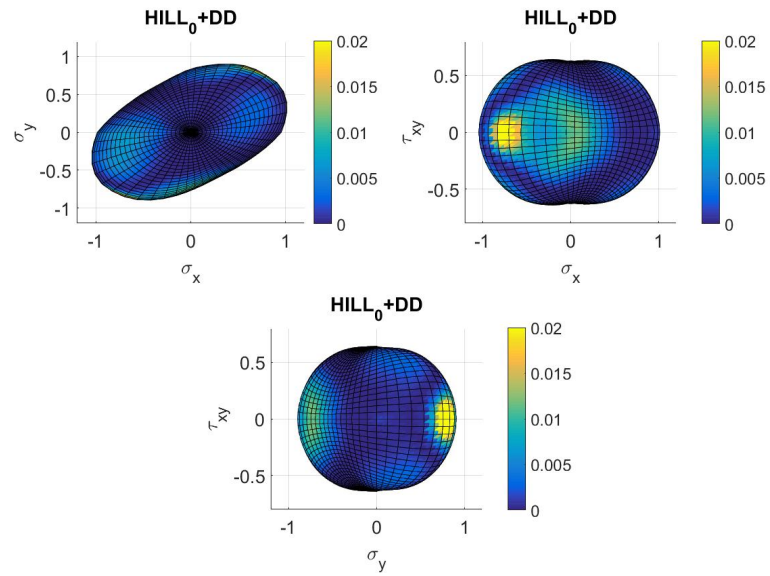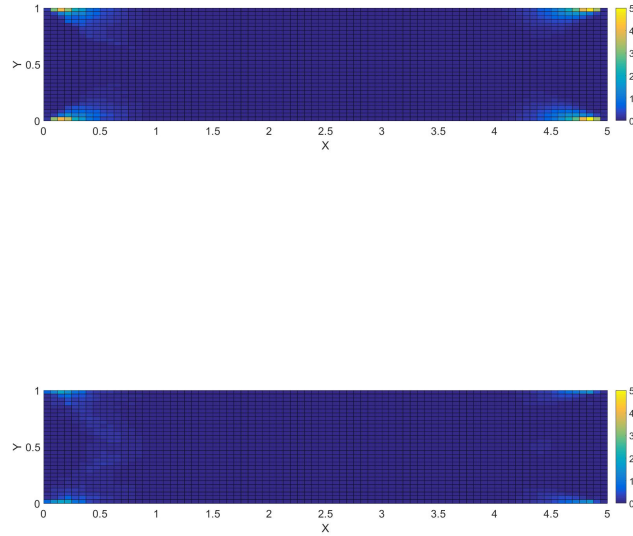

Figure 3.34 – Different views of $\mathcal{H}_2 + DD$ plastic yield function. Colours are mismatch between the $\mathcal{H}_2 + DD$ criterion and the Barlat Yld2004-18p. $E_{\mathcal{F}_Y}^{\mathcal{H}_2 + DD} = 4.63$.

Figure 3.35 – Cumulated strain error between Barlat's and Hill in top and between Barlat's and Hill plus data correction in bottom.

Finally, a final simulation of a bar with the two right and left extremes campled in which a uniform vertical negative distributed load in both bottom and top sides of the bar is acting. Fig. (3.35) shows the cumulated strain error between Barlat's and Hill's yield functions in the top. Bottom figure shows the error between Barlat's and Hill's plus data driven correction yield functions. As it can be seen, the error in the strain field is reduced when considering the correction. However, this error does not go to zero since the correction does not reproduce perfectly well Barlat's criterion.

3.2.2.3 Conclusions

In the light of the results, data driven correction importance is higher when the model is less accurate since the very beginning. An alternative route based on data completion techniques in order to circumvent the problem of minimizing a function in the high dimensional space is proposed in this work. The sPGD algorithm seems to be a suitable candidate to generate response surfaces within this model correction framework, where both non-structured data-set and high dimensional spaces arise as soon as a rich correction spaces are considered. Once the discrepancy error response surface is built, its minimum provides a good candidate to enrich the existing model, the same way it will be provided by the minimum of a standard optimization algorithm.
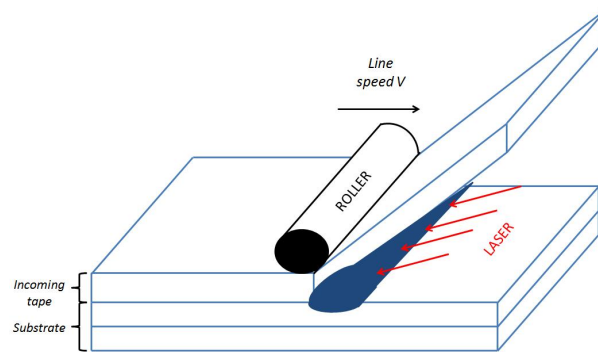
Figure 3.36 – Sketch of the automated tape placement (ATP).

3.2.3 Prediction of composites behavior undergoing an ATP process through data-mining

Consolidation of preimpregnated preforms, e.g. sheets, tapes, constitutes one of the main composite forming processes for manufacturing structural parts. Among the vast variety of technological solutions available, the automated tape placement –ATP– is gathering a lot of interest due to its versatility and its capacity to avoid the use of autoclave.

Fig. 3.36 ilustrates how a tape is placed and progressively welded on the substrate consisting of the tapes previously laid-up. The welding of two thermoplastic layers is fulfilled whenever specific physical conditions are achieved: an almost perfect contact between layers (intimate contact) and a sufficient temperature enabling molecular diffusion within the process time window, while avoiding thermal degradation. Because the characteristic low thermal conductivity of usual resins, an intense local heating is usually considered (laser, gas torches, ...) in conjunction with a local pressure applied by the consolidation roller moving with the heating head. In this process particularly, heat plays a double role, first it enhances molecular mobility and secondly, the decrease of the material viscosity with the temperature increase, facilitates the squeeze flow of the heated asperities located on the ply surfaces under the compression applied by the consolidation roller.

Subsubsection 3.2.3.1 proposes a set of surface descriptors able to characterize the evolution of the degree of intimate contact during processing. That knowledge is crucial for the online process control in order to maximize both productivity and part quality. Ideally, a response surface linking the geometrical descriptors together with the consolidation degree would be interesting from an industrial point of view. For that purpose we discuss different surface descriptors and different meta-modeling strategies based on either regression trees or sPGD algorithm presented in section 3.1. Details on the consolidation simulation will be skipped since the main aim of this section is to focus on sPGD applications. Therefore, the time required to consolidate a given material surface will be given as an input to the meta-modeling algorithms.

Once a simulation of the ATP process is performed on a set of real surfaces, the output of interest, the consolidation time, time require to fulfill total contact between two adyacent layer, is intended to be expressed from few geometrical parameters describing the surface with respect to the physics considered (here the heating and squeeze flow). Then, both surrogate modeling approaches, based on either regression trees or sPGD, are then compared and validated is subsections, 3.2.3.2 and 3.2.3.3, respectively.

3.2.3.1 Surface descriptors

Roughness parameters were considered in many works [Dagnall 2014, Bhushan 2001, Torquato 2002]. ISO 16610-21 provides an insight about the way of addressing surface analysis by differentiating waviness (associated to the macro-scale) and roughness (related to the micro-scale). A gaussian filter (as ISO 16610-21 recommends) is considered for that purpose. Thus, if Z represents the surface height, the ones related to the waviness and roughness are denoted respectively by Z_w and Z_r, according to

$$Z = Z_w + Z_r. \tag{3.23}$$

Figure 3.37 depicts for six profiles extracted from a real prepreg by using a profilometer, their macro-scale waviness and micro-scale roughness.

Inspired by the fact that most of diffusive transport phenomena are induced by curvature, we consider in the present work the curvature of the surface as a potential descriptor of the surface with respect to the crushing time where the squeezing flow occurs. Intuitively, we can associate low curvature as large plateau of the surface topography. Squeezing such a large plateau under the effect of the applied pressure is difficult since all the the material must flow. On the contrary, a surface exhibiting high curvature can be visualized as having many thin pics, that easily flows to fill the neighbor valleys when compressed, as sketched in Fig. 3.38.

For better quantifying the size effect we consider at time t the rectangular domain representing an asperity $\Omega_t = [-L_t/2, L_t/2] \times [-H_t/2, H_t/2]$ occupied by a Newtonian fluid characterized by a viscosity η, with, $L_t \gg H_t$. This element results from the compression of the initial rectangular element $\Omega_0 = [-L_0/2, L_0/2] \times [-H_0/2, H_0/2]$, with $H_t L_t = H_0 L_0$ ensuring the fluid incompressibility. We assume that on the upper boundary $z = H_t/2$ a normal force applies F. The associated compression rate will be noted by W (see Fig. 3.39). Assuming lubrication theory hypothesis, the applied force and the compression rate verify the relation,

$$F = \frac{W \eta L_t^3}{H_t^3}, \tag{3.24}$$

where F is the force per unit depth. This proves that for the same compression rate and thickness evolution, the higher is the asperity length L_t the more intense force F must be applied, or equivalently, the same force squeezes faster thin asperities.

Thus, curvature seems to be a key parameter describing the surface and with a clear physical content in what concerns the squeezing flow leading to intimate contact. However, that curvature can be defined, in real surfaces, and more particularly in pre-impregnated composites, at different scales. Thus, in what follows and inspired by ISO 16610-21, we will consider the curvature associated with both the macroscopic waviness and the microscopic roughness, and it is expected the former, associated with larger space scales, controlling the slow squeezing dynamics of the process. Intuitively, optimal consolidation is expected to be associated with negligible macro-curvature and very high micro-curvature.

Appart from the curvature, other potential geometrical parameters are also considered: the volume of fluid as well as the slope (first derivative of the surface profile) at both the macro and micro scales. Those parameters are expressed from:

- Macro-slope: Z_w', with $\bullet'$ the first derivative.

- Macro-curvature defined from the standard deviation of the Z_w'' according to

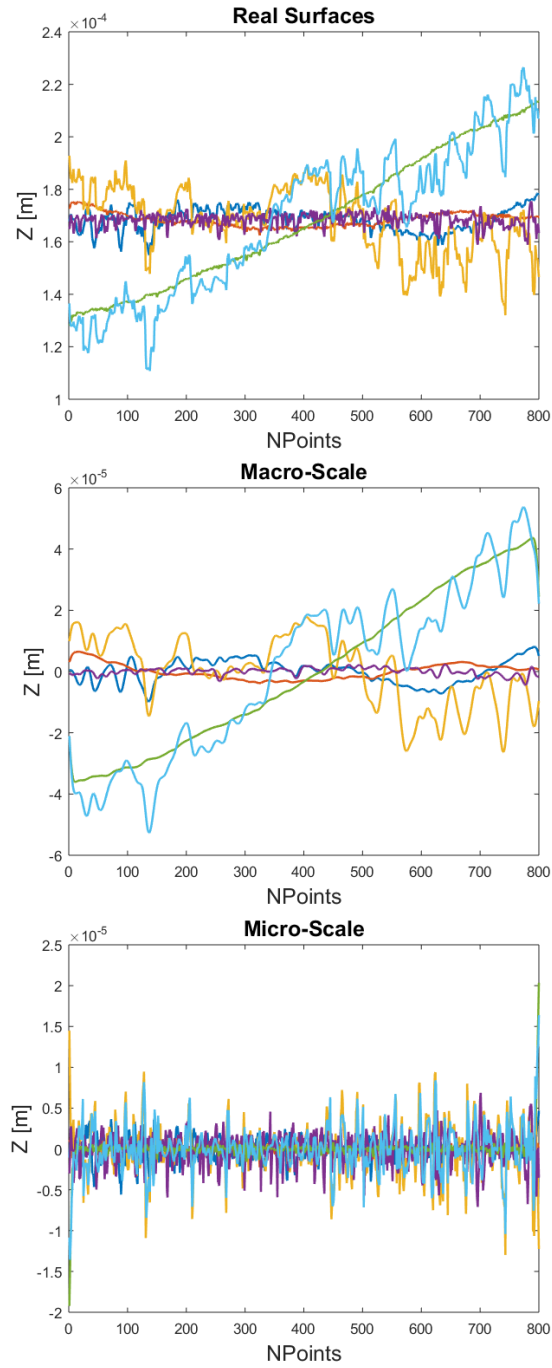$$\kappa_w = \left| \frac{\sigma(Z_w'')}{\sigma(Z_w)} \right|, \tag{3.25}$$

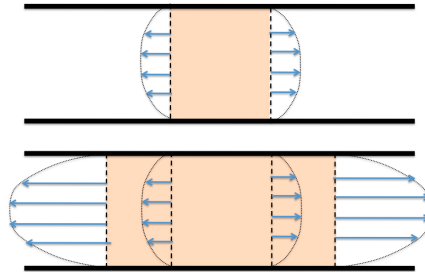Figure 3.37 – Differentiating waviness and roughness of six real surfaces

Figure 3.38 – Squeezing rectangular fluid volumes representing asperities of different length.
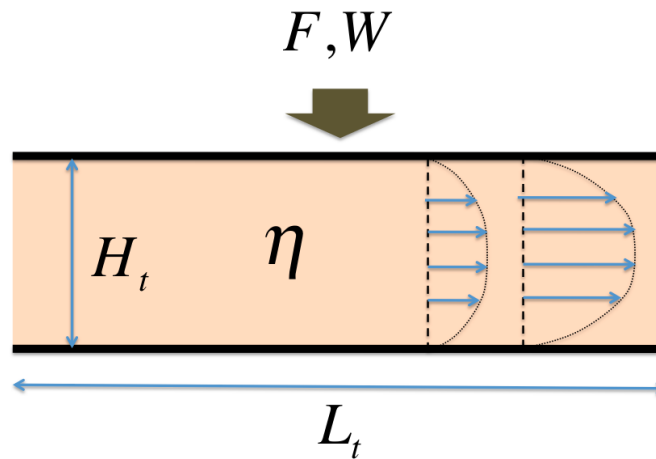

Figure 3.39 – Squeezing a rectangular fluid volume representing an asperity.

where σ refers to the standard deviation, and $\bullet''$ the second derivative.

- Micro-slope: Z_r'.
- Micro-curvature

$$\kappa_r = \left| \frac{\sigma(Z_r'')}{\sigma(Z_r)} \right|. \tag{3.26}$$

- The normalized volume of fluid

$$A = \frac{\int Z dy}{\mathcal{A}_{max}}, \tag{3.27}$$

with $\mathcal{A}_{max}$ the maximum value of A for the different considered surfaces.

In what follows, and from these tentative set of parameters, we are trying to express the dependency of the crushing time with respect to the surface geometry, the last described from the above parameters. For that purpose, first we consider data-mining techniques, in particular "decision trees" (freely available in several softwares), due to its ability to classify and to generate regressions. Thus, on one hand we expect to understand the relevance of these parameters, their relative significance, and on the other hand, after validating its relevance and completeness, estimate the crushing time for any other surface described from another value of the set of geometrical parameters.

An alternative route to generate such a response surface relating the crushing time and a set of geometrical parameters easily accessible and allowing a non ambiguous determination of the output of interest (i.e. the same parameters cannot be associated with different values of the output), consists of defining a multidimensional polynomial relating both, inputs (geometry) and output (consolidation indicator, here the crushing time). However multidimensional interpolation or approximation can suffer of the so-called curse of dimensionality.

In the sequel, two different tools are used to generate the response surface of the crushing time as a function of the geometrical parameters, namely, decision trees and sparse Proper Generalized Decomposition.

3.2.3.2 Consolidation modeling by decision trees based regression

We consider a set of $M = 1359$ surface profiles extracted from 16 real different preimpregnated composite parts, and run an ATP simulation on each profile to obtain the time evolution of the intimate contact. The analyzed profiles and the time evolution of the intimate contact are depicted in Fig. 3.40.

The data-set consisting of the crushing time (the one at which an almost perfect contact is attained) and the 5 geometrical parameters previously introduced (micro and macro curvatures, micro and macro slopes and volume of material) was used to create the searched regression relating the crushing time with the surface geometrical parameters. Figure 3.41 depicts the obtained results where it can be noticed that as expected the macro-curvature appears as the most relevant parameter for explaining processability. The obtained regression has a reliability of $91, 43\%$, that correspond to the expected success when predicting the target.

To validate the procedure, a tree-based regression was performed by using a part of the available data (training data), and then comparing the predictions with the data for the remaining surfaces. In order to quantify the predictability capabilities we define the error ϵ

$$\epsilon = \sqrt{\sum_{i \in \mathbb{S}} \frac{\left(t_i^{ref} - t_i^{est}\right)^2}{\left(t_i^{ref}\right)^2}}, \tag{3.28}$$
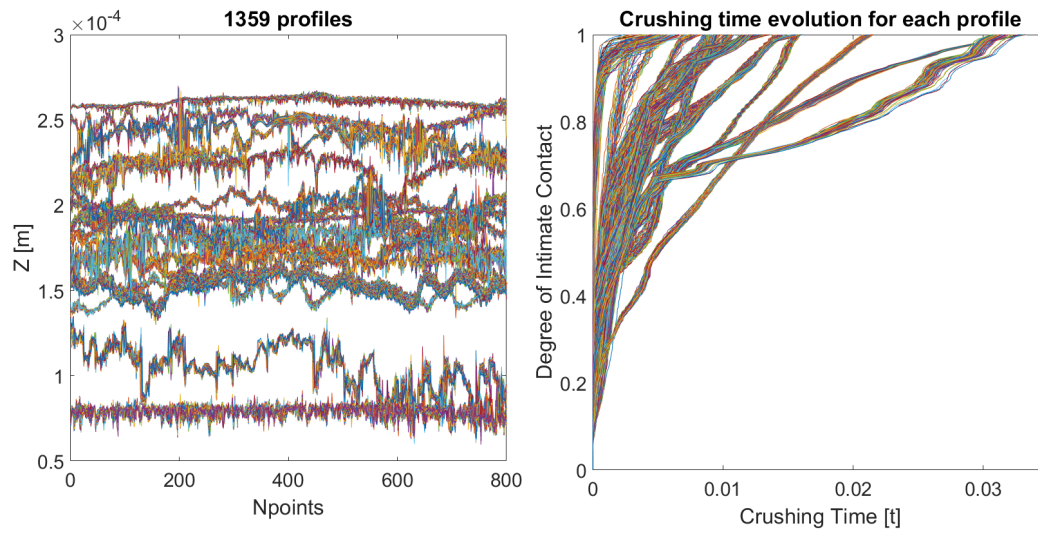
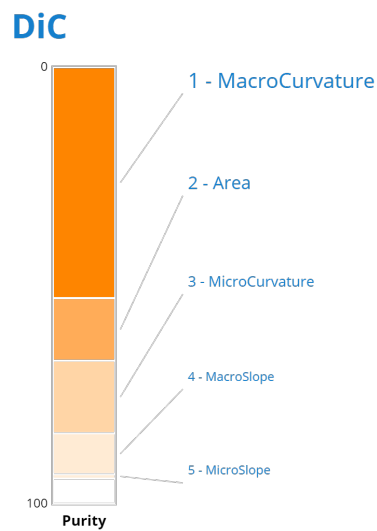Figure 3.40 – Analyzed surfaces and their crushing time evolution.

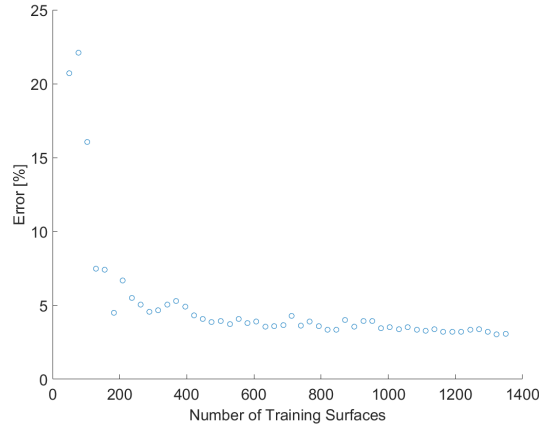Figure 3.41 – Parameter weights explaining the degree of intimate contact based on decision tree

Figure 3.42 – Error versus number of surfaces considered for defining the tree-based regression

where $\mathbb{S}$ is the set of surfaces that were not involved in the regression model construction, and t_i^{ref} and t_i^{est} are respectively the reference crushing time (obtained by simulating numerically the squeezing process) and the one estimated by using the repression model, both related to each surface $i \in \mathbb{S}$.

Figure 3.42 shows the evolution of error ϵ for estimating the crushing time depending on the number of surfaces used to build the regression tree. As expected, this error decreases by increasing the number of surfaces used for the model training. It can be noticed that the error stagnates (of about 4%) when considering more than 40% of the available surfaces in the regression construction.

3.2.3.3 Consolidation modeling by sPGD based regression

This part aims at comparing the tree-based regression with the one obtained by using the sparse Proper Generalized Decomposition –sPGD– previously introduced in 3.1. The main advantage of the last is its versatility and moreover, it becomes fully combinatorial with respect to tree-based regression in which the curse of dimensionality was circumvented by defining a sequence of the different branches constituting the tree.

Manifold learning allows clustering surfaces with respect to the output of interest, here the crushing time, as Fig. 3.43 reveals, where color is related to the magnitude of the crushing time. For the sake of visualization we considered just three of the five involved parameters for the representation.

Now, the sPGD looks for approximating the crushing time $\mathcal{C}_t$ depending on the five selected parameters: κ_w, κ_r, Z'_w, Z'_r & A; in the separated form

$$\mathcal{C}_t(\kappa_w, \kappa_r, Z'_w, Z'_r, A) \approx \sum_{i=1}^{c} F_i^1(\kappa_w) \cdot F_i^2(\kappa_r) \cdot F_i^3(Z'_w) \cdot F_i^4(Z'_r) \cdot F_i^5(A), \qquad (3.29)$$

where the so-called modes F_i^k, depending on the different parameters, are approximated by using 1D-Kriging interpolants.

The solution obtained by using the sPGD is then compared to the one resulting from tree-based regression, using the same error indicator ϵ previously introduced in Eq. (3.28) when considering 50% of the surfaces for constructing the model.
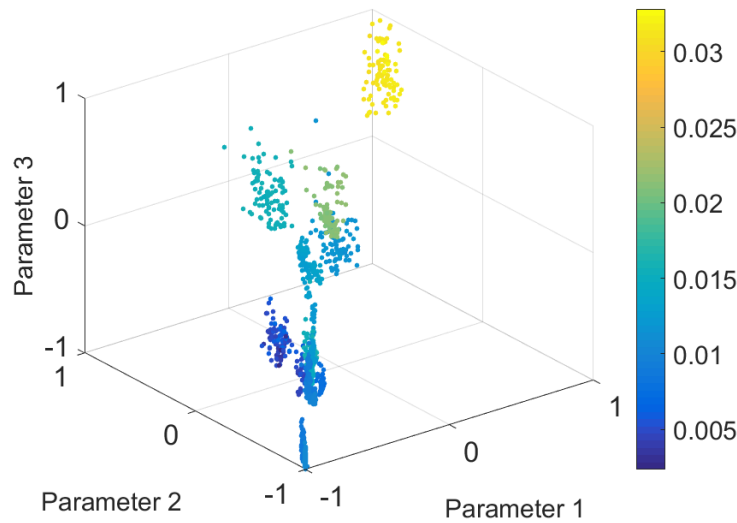
Figure 3.43 – Surface clustering and multidimensional visualization in a 3D space.

Fig. 3.44 depicts the evolution in the predictions obtained from the sPGD depending on the number of modes considered in the PGD separated representation. Stars represent the error for the surfaces that served to construct the approximation, that is not zero because the considered approximation (using 12 kriging control point in each direction) is unable to fulfill a perfect interpolation of all the training points. Circles are used when considering all the surfaces, and as expected it is a bit higher than the previous one (that only considered the surfaces used for constructing the nonlinear sPGD-based regression). However, the error gap is quite small, and as in the case of the tree-based regression, the error related to the sPGD approach stagnates around 5% error.

Now, taking into account that surfaces exhibit a noticeable clustering, as previously discussed and seen in figure 3.43, we could define a sPGD operating at each cluster level. First using the k-means technique based on the macro-curvature that is expected having the more important effect on the intimate contact evolution, surfaces were grouped in the three clusters depicted in Fig. 3.45.

When applying local sPGD approximation at the level of each cluster the error reduces significantly as Fig. 3.46 proves. Indeed, each cluster presents a weaker non-linearity compared to the global problem, thus, the task entrusted to sPGD algorithm for each cluster is easier than the global one.

3.2.3.4 Conclusions

The main aim of the present work was to efficiently predict composite processability by evaluating the intimate contact from few geometrical parameters easily accessible. From a data-base consisting of thousands surface profiles, consolidation was simulated by solving numerically, by using the PGD, the heating and squeeze flow coupled problem. Curvature was expected being of major relevance in the process. Data-assimilation techniques allowed to confirm this hypothesis and also to create a regression model relating the output of interest (the crushing time at which the contact becomes almost perfect) with a set of five geometrical parameters. Tree-based regression was considered for that purpose, and the
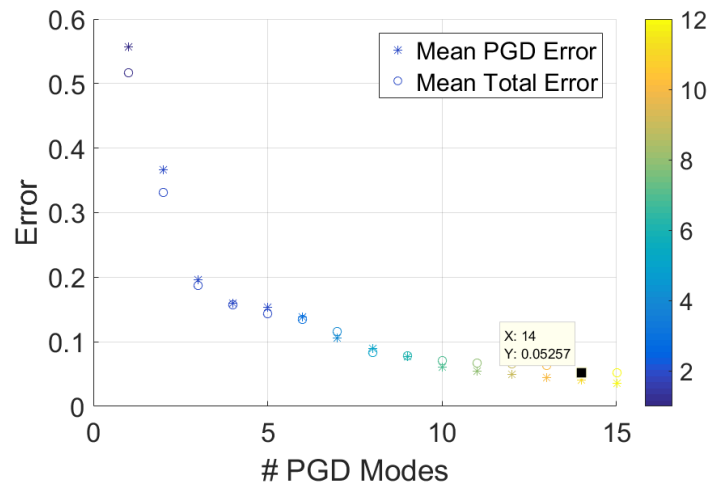
Figure 3.44 – Error when using sPGD-based approximation with 50% of the surfaces for defining the regression model.
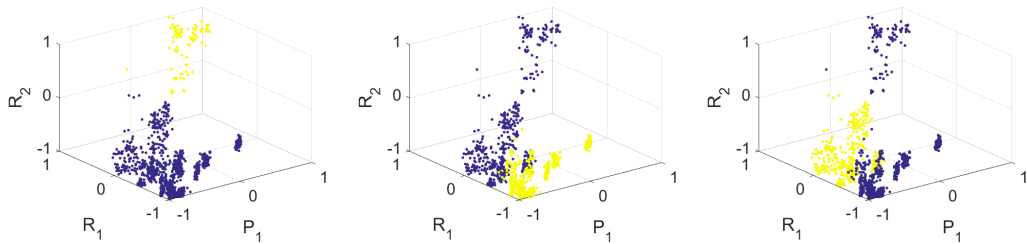

Figure 3.45 – Surface clusters in which the local sPGD applies. Different clusters are based on the yellow points.
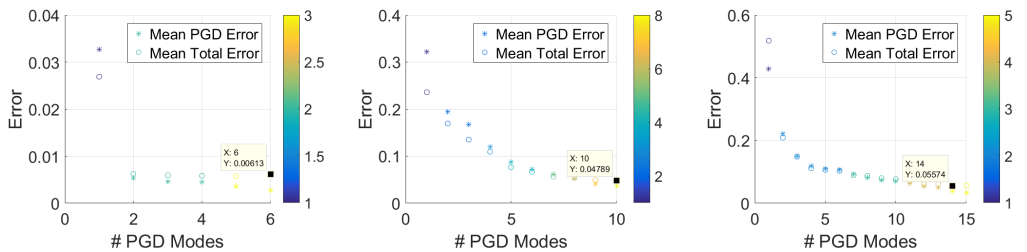

Figure 3.46 – Error evolution in each cluster for estimating the data that served to construct the model (Mean PGD Error) and the total set of data (Mean Total Error)

evaluated error between the estimated crushing time and the one simulated differ of about 5% proving the potential of the proposed approach.

Another route with greater physical significance was implemented, the one consisting of a nonlinear regression based on a multidimensional polynomial approximation. In order to address the two main issues encountered, first the sparse available data in the parametric space and second its multidimensional nature, we used a sparse variant of the PGD-based approximation that combined with an adaptive kriging approximation in the different parametric dimensions allowed to achieve accuracies of the same order of magnitude than the ones obtained by using tree-based regression. Moreover, the accuracy was enhanced when considering local sPGD-based nonlinear regression at the level of surface clusters, the last constructed by making use of a classification strategy (e.g. k-means) with respect to the output of interest (here the crushing time).

3.3 Applications of Compressed Sensing in Computational Mechanics

Compressed sensing is a signal compression technique with very remarkable properties. Among them, maybe the most remarkable one is its ability of overcoming the Shannon-Nyquist sampling theorem. In other words, it is able to reconstruct a signal at less than $2Q$ samplings per second, where Q stands for the highest frequency content of the signal.

This property has, however, important applications in the field of computational mechanics, as we analyze in this section. We consider a wide variety of applications, such as model order reduction, manifold learning, data-driven applications and nonlinear dimensionality reduction. Examples are provided for all of them that show the potentialities of compressed sensing in terms of data completion techniques in the field of computational mechanics.

In this work, we address the problem of the reconstruction of a parametric solution from a coarse sampling, but from a different perspective. Compressive sensing provides a solid framework for performing random samplings. There is a vast literature on compressive sensing, extensively used in data and image analysis (see [Kutz 2013, Mangan *et al.* 2016, Kaiser *et al.* 2018] and the references therein). It has recently attracted the interest of the modeling and simulation scientific communities to works like [Brunton *et al.* 2016]. In subsection 3.3.1, we first revisit the main concepts related to compressive sensing. Subsection 3.3.2 applies this methodology to generate response surfaces of parametric models. Subsections 3.3.3 and 3.3.4 presents and discusses some other applications related to model order reduction and transient problems, respectively.

3.3.1 Overview of compressed sensing

Most of nonlinear dimensionality reduction techniques consider least-squares fitting of the data. However, compressed sensing is based in the use of the L^1 norm instead. As described in [Kutz 2013], there is a subtle link between sparsity and the use of the L^1 norm. When considering curve fitting, the use of standard L^2 norms magnifies the importance of outlying points because of the squared norm. The impact of these outlying points in the fitted curve can be significant.

In the same spirit, the solution of underdetermined algebraic systems is a tricky issue because they represent an infinite number of solutions. As illustrated in [Kutz 2013], the use of the pseudo-inverse produces a fully populated solution vector whereas when considering the "Matlab" backslash, the obtained solution contains many zero entries, so that it results

to be sparse. When solving the problem with L^2 and L^1 optimizations (trying to obtain the minimum norm solution), the former becomes much less sparse than the last. In the case of overdetermined systems the same tendencies can be observed.

Thus, from a purely engineering viewpoint, L^1-norm can be associated to sparsity. For this reason the L^1 norm was considered as an appealing candidate for addressing signal reconstruction. It is able to overcome the Nyquist-Shannon sampling theory that states that for recovering a signal, one must sample at twice the rate of the highest frequency involved in the signal.

Let us consider a vector $\mathbf{f}$, in the usual space or time domains, and its counterpart in a domain in which it should accept a sparse representation, i.e., its vector counterpart $\mathbf{c}$ contains many zeros. These spaces are in general the ones related to frequency (Fourier or discrete cosines transforms) or the ones related to multi-resolution wavelets, among many other possible choices. We denote by $\mathbf{T}$ the matrix making possible this discrete transformation, i.e.,

$$\mathbf{Tc} = \mathbf{f}. \tag{3.30}$$

Since vector $\mathbf{c}$ is expected to have many zero entries (as soon as it corresponds, by assumption, to a space in which the signal becomes sparse), one could expect that its expression could be determined by employing only some rows of matrix $\mathbf{T}$ and vector $\mathbf{f}$. This implies solving the resulting underdetermined system making use of a L^1-norm based optimization.

The choice of such rows can be made in different ways. However, the most usual one consists of a random selection. From a matrix perspective, such extraction simply consists of defining a diagonal matrix, with unit entries at the rows we want to extract. If the set of rows to be extracted is denoted by $\mathcal{S}$, the extraction matrix $\mathbf{E}$ is defined from

$$\begin{cases} E_{ii} = 1 & \text{if } i \in \mathcal{S}, \\ E_{ij} = 0 & \text{otherwise.} \end{cases}$$

Rows containing only zeros are then eliminated from the matrix, thus generating a rectangular one, here denoted as $\mathbf{E}$.

The solution of problem (3.30) can thus be approximated by that of the underdetermined system

$$\mathbf{ETc} = \mathbf{Ef}, \tag{3.31}$$

by using a L^1-norm based optimization.

In sum, the two main ingredients to succeed whenever using a compresive sensing technique are: (i) the use of an adequate space in which the solution of the problem at hand is expected to exhibit sparsity, and (ii) the solution of the underdetermined problem by using a L^1 norm.

Compressed sensing is at the origin os the so-called "single pixel camera". In it, instead of acquiring the global image information, i.e., a pixel vector $\mathbf{f}$, to be then compressed, only a few of its entries are acquired, namely $\mathbf{Ef}$. As soon as vector $\mathbf{c}$ is calculated by solving Eq. (3.31), the whole solution can be reconstructed from Eq. (3.30).

In the sequel, to solve the system of equations (3.31) we have preferred to employ the least absolute shrinkage and selection operator, LASSO, method [Tibshirani 1996]. Just as a recall, LASSO solves a minimization problem that involves an L2- minimization of the system of equations plus a penalty term involving a L1-norm of the unknown field,

$$\min_{\mathbf{c}}(||\mathbf{ETc} - \mathbf{Ef}||_{L2}^2 + \lambda||\mathbf{c}||_{L1}).$$

This simple modification of the original compressed sensing procedure has revealed to avoid some of the numerical problems associated with the solution of an under-constrained system, providing a new coordinate system such that the solution is sparse. Its numerical performance is analyzed in subsubsection 3.3.1.1 below.

3.3.1.1   Numerical Performance of the LASSO scheme

The functional that the LASSO algorithm seeks to minimize does not have a closed-form solution. Hence, an iterative algorithm has to be used in order to find the minimum of the functional, i.e. a steepest descent method. When using LASSO, there is a numerical parameter called the relative tolerance, which controls the maximum admisible difference between two iterations of the steepest descent method. If the difference between two iterations is smaller than the relative tolerance, the iterative algorithm stops. Generally speaking, the smaller the relative tolerance, the higher number of iterations are required, but the solution will be more accurate. Fig. 3.47 shows the relative tolerance of LASSO method versus solution error in logarithmic scale. The colors of the points represent the time, in seconds, required to solve the underdetermined system by means of LASSO. As it can be noticed, the lower the relative tolerance, the lower error in the solution is obtained. Yet, the time to solve the system increases. In this particular case, a reconstruction of the temporal evolution of a vector $\mathbf{u}(t)$ that evolves with time is intended to be reconstructed from few snapshots. The number of spatial snapshots $\mathbf{u}^i$ to build the LASSO system was 10 out of 75 total time steps. The solution error is measured by

$$\epsilon_U = \frac{1}{N}||\hat{\mathbf{u}} - \hat{\mathbf{u}}_R||_{L2},$$

where $\hat{\mathbf{u}}_R$ is the reference solution, which is assumed to be the one of the usual time marching approach. N represents the number of total unknowns.

At the end of the day, a compromise between the solution error and computational cost should be accomplished. The quality of the reconstructed solution depends also on the number of snapshots. Fig. 3.48 shows the number of snapshots versus solution error in logarithmic scale. The legend represents the time required to solve the LASSO minimization. The relative tolerance is set to $0.8e - 5$. Obviously, the more snapshots $\mathbf{u}^i$ we consider, the more accurate is the solution and less time is required to solve the system of equations.

In our cases, a relative tolerance of $1e-5$ provides good results. If the relative tolerance is set to $1e - 6$, it will provide even better results in terms of error, but it will take more time to minimize the functional. In this particular case, the Discrete Cosinus Transform (DCT) provided good results as plenty of coefficients in the new basis are equal to zero. Fig. 3.49 shows the DCT coefficients solving the entire system of equations (red) and the ones obtained after minimizing LASSO (blue) with a relative tolerance of $1e - 4$. As it can be noticed, there are some differences between the red and blue curves, meaning that the iterative algorithm has finished before reaching the proper minimum of the functional. Fig. 3.50 shows how imposing a smaller tolerance will provide better results. Of course, the price to pay is that the LASSO minimization takes more time.

3.3.2   Response surface generation by means of compressed sensing

As discussed in the introduction, parametric solutions of the type $u(\mathbf{x}, t, \boldsymbol{\mu})$—where $\boldsymbol{\mu}$ represents the set of parameters in the problem—obtained by the (off-line) application of PGD techniques are extremely valuable for conducting (on-line) real-time simulations as well as optimization, inverse analysis, simulation-based control and uncertainty propagation under
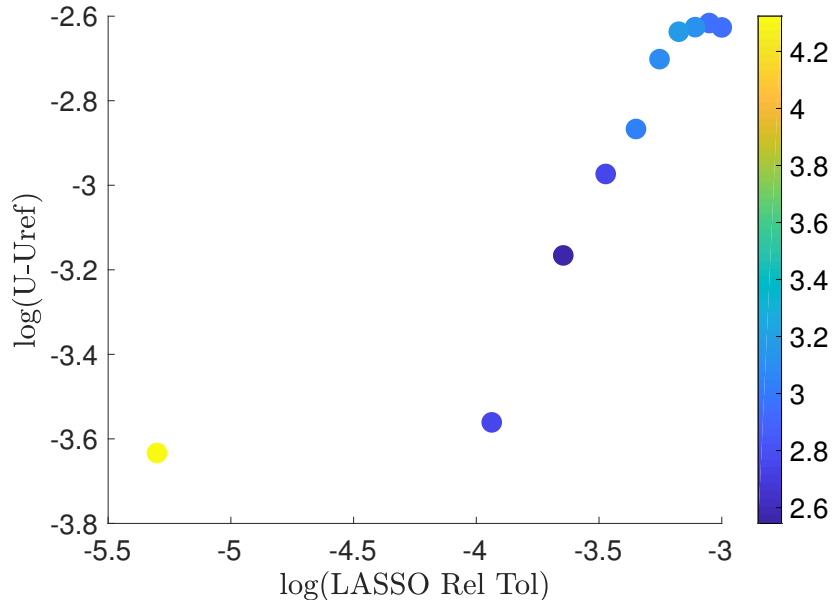
Figure 3.47 – Relative Tolerance of LASSO method versus solution error in logarithmic scale. The legend represent the time, in seconds, required to solve the LASSO minimization. 10 out 75 time steps are sampled.
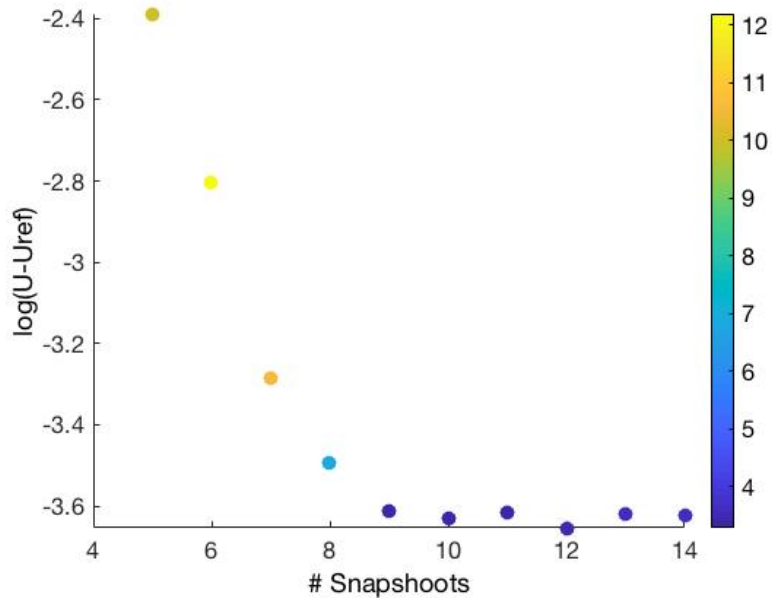

Figure 3.48 – Number of snapshots versus solution error in logarithmic scale. The legend represents time, in seconds, required to solve the LASSO minimization. Relative tolerance set to $1e - 5$.
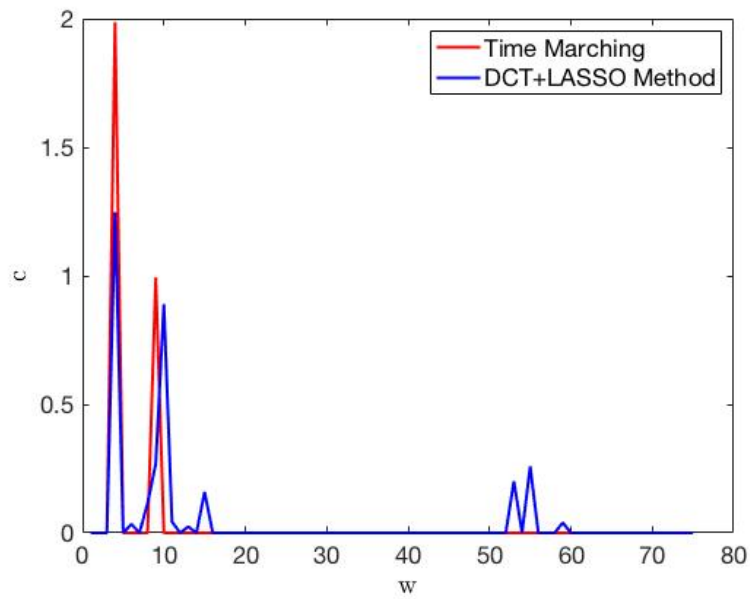
Figure 3.49 – DCT coefficients at a given spatial node with a bad tolerance in LASSO. (Relative tolerance 1e-4).
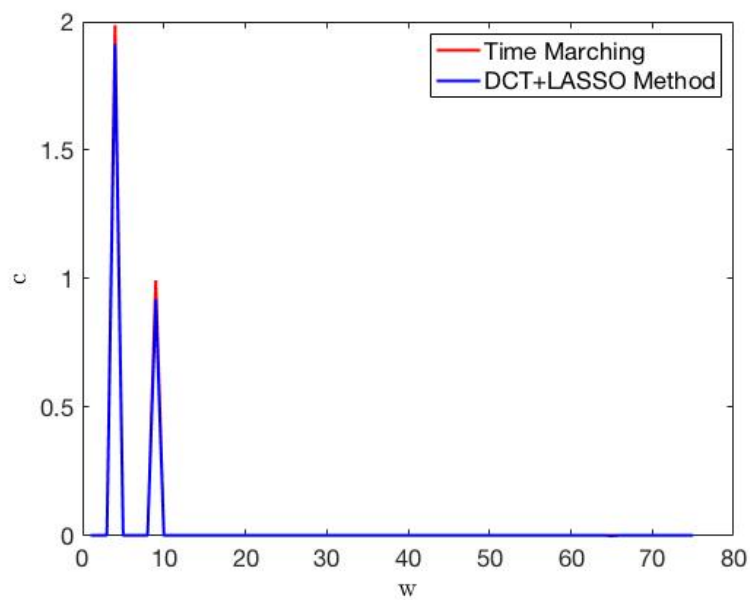
Figure 3.50 – DCT coefficients at a given spatial node equation with a good tolerance in LASSO. (Relative tolerance 0.8e-5).
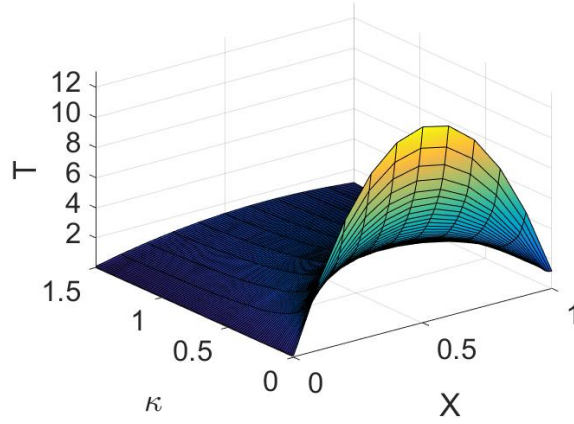
Figure 3.51 – Parametric solution $u(x, \kappa)$ of Eq. (3.32).

real-time constraints [Chinesta & Cueto 2014, Cueto *et al.* 2016]. However, the standard PGD constructor is strongly invasive with respect to the use of commercial simulation codes. To circumvent or, at least, alleviate such a constraint, sampling of the parametric space and a subsequent interpolation of these samples could be an alternative route. However, by making it in the original space the sampling becomes sometimes too dense to capture the richness or as consequence of the Nyquist theorem. As just discussed in the previous section, compressed sensing by-passes such difficulties in many cases.

In order to illustrate the proposed procedure, we consider the parametric heat equation

$$\kappa \frac{\partial^2 u}{\partial x^2} = s, \quad \text{in } \Omega_x = (0, L = 1), \tag{3.32}$$

with κ the thermal conductivity, $s = 1$ the source term (assumed constant in the space domain), and with boundary conditions $u(x = 0) = 0$ and $u(x = L) = 1$.

We are interested in solving this thermal model for any thermal conductivity $\kappa \in \Omega_\kappa = [0.1, 1.5]$. Solving it using the standard PGD approach (for an in-deep discussion of this problem the interested reader should consult [Chinesta & Cueto 2014, Cueto *et al.* 2016]) we obtain the solution depicted in Fig. 3.51. It reveals that, by increasing the conductivity, the solution becomes flatter, since the generated heat can easily reach the domain boundaries $x = 0$ and $x = L$ leaving the domain. Lower conductivities imply higher temperatures because of the difficulty of evacuating the produced heat.

In order to show the potential of compressed sensing, we consider $K = 100$ coordinates along Ω_κ. From them, we randomly select $K_r = 10$ samples, defining the sampling set $\mathcal{S} = \{\kappa_1, \ldots, \kappa_{K_r}\}$. At these particular *locations*, Eq. (3.32) is solved by using standard finite differences or finite elements. These discrete solutions, consisting of vectors containing nodal temperatures for each choice of the thermal conductivity are denoted by $\mathbf{u}^k$, $k = 1, \ldots, K_r$.

Considering now different nodes in the spatial mesh associated to Ω_x, x_i, we define vectors $\mathbf{f}_i$ whose j-th entry reads

$$f_{ij} = u^{\kappa_j}(x_i).$$

Vector $\mathbf{f}_i$ contains $K - K_r$ unknown entries, those related to thermal conductivities not in the sampling set $\mathcal{S}$. However this does not constitute a problem, since the corresponding rows are not affected by the extraction matrix $\mathbf{E}$. Only the rows with known entries will be extracted.
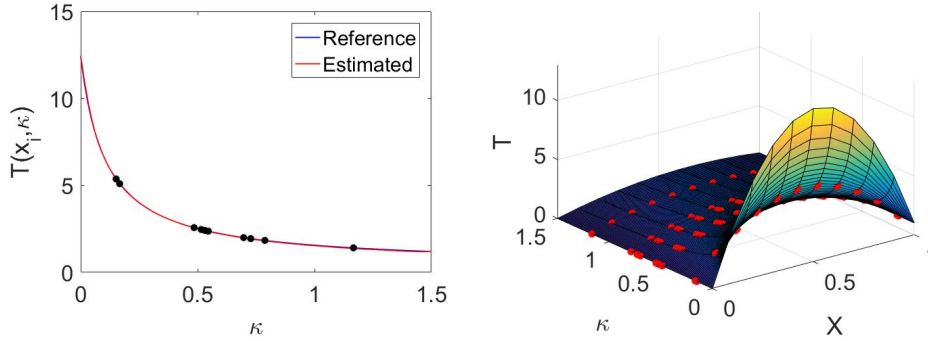
Figure 3.52 – Reference versus compressed sensing based reconstruction of the nodal solution evolution with the model parameter. Left: Solution at a given nodal position. Black nodes represent the sampling points. These are also indicated on the right.

At this point, by using a wavelet representation to define the transformation matrix $\mathbf{T}$—in particular, a biorthogonal 3.1 one—, the parametric model at node x_i will read

$$\mathbf{Tc}_i = \mathbf{f}_i, \tag{3.33}$$

or, by extracting the selected entries,

$$\mathbf{ETc}_i = \mathbf{Ef}_i. \tag{3.34}$$

Solving the underdetermined problem using a L^1-norm, we obtain $\mathbf{c}_i$, from which the nodal parametric equation can be reconstructed everywhere from

$$\mathbf{f}_i = \mathbf{Tc}_i. \tag{3.35}$$

Figure 3.52 compares the reference and reconstructed solutions at a particular node. We deliberately chose one located in the center of the domain in order to involve large gradients. In this figure the sampling nodes are also depicted.

We thus see how by sampling only the ten percent of all parametric nodal positions we obtain a remarkable accuracy in the reconstruction of the thermal field. Notably, this strategy allows the use of commercial software to obtain response surface-like solutions to parametric models and constitutes an alternative and valuable constructor of meta-models. It avoids the typical oscillations that polynomial approximations provoke when the sampling points are chosen randomly and do not correspond with the Gauss-Lobatto points. In that sense, compressed sensing produces solutions closer to the ones associated with the use of Kriging, that avoids large oscillations thanks to its statistical nature.

3.3.2.1 Constitutive Manifold Learning

Compressed sensing could also play a fundamental role in data-driven simulations. In section 2.1, we proved that elastic data-driven simulations can be performed as soon as stress/strain couples are given. However, as previously argued in section 2.2, prior to proceed with the calculations summarized above, one must accomplish the construction of the so-called constitutive manifold. Therein, we considered an inverse approach that assumed a tentative constitutive manifold. We proved in the capability of such a method to identify

the constitutive manifold associated to nonlinear elasticity. However, its generalization to more complex behaviors—like those involving internal variables—seems technically complex [Ibañez *et al.* 2018] [Gonzalez *et al.* 2018a].

One possible route to explore consists in making use of well-experienced experimental methodologies. These were often developed for calibrating constitutive equations by testing coupons subjected to simple stress states. These tests are very well understood and offer valuable information for calibrating complex constitutive equations. However, by restricting to them, the constitutive manifold remains mostly unexplored, since too sparse information is accessible.

It is at this point that compressed sensing seems to offer a valuable opportunity. Indeed, if the constitutive manifold is viewed as a sort of image of the phase space of the material, a small quantity of data points could be enough to determine the whole manifold. A parallelism could be established with the so-called single pixel camera, following the rationale described in [Kutz 2013].

To evaluate the performance of such a procedure, we consider a hypothetical nonlinear plane-stress elastic behavior, that in Voigt notation reads

$$\mathbf{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \tag{3.36}$$

with the elastic coefficients given by

$$\begin{cases} E = E_0 + E_1 \mathrm{Tr}(\boldsymbol{\varepsilon}) \\ \nu = \nu_0 + \nu_1 \mathrm{Tr}(\boldsymbol{\varepsilon}) \end{cases},$$

with E_0, E_1, ν_0 and ν_1 positive constants, and where $\mathrm{Tr}(\bullet)$ refers to the trace operator acting on tensor $\bullet$. In the numerical example discussed below the material coefficients were selected as $E_0 = 10$, $\nu_0 = 0.1$, $E_1 = 10$ and $\nu_1 = 0.1$.

We considered different strain couples $(\varepsilon_{xx}, \varepsilon_{yy})$ and determine from the constitutive Eq. (3.36) the associated stress couples $(\sigma_{xx}, \sigma_{yy})$. Fig. 3.53 depicts the reference solutions and the considered points (in red) that served to reconstruct the approximated manifold from the compressed sensing rationale. We do not considered the off-diagonal components because they define a one-dimensional manifold that is quite simple to approximate, as proved in the previous section.

It can be noticed that, despite the small number of sampling points, the reconstructed constitutive manifold reproduces accurately the reference solution, thus constituting an alternative route to perform data completion whenever the experimental data set is reduced. Obviously, an advantage of the methodology here described is the possibility to update the reconstructed solution as soon as new data-points are available from testing facilities able to explore new regions of the constitutive manifold. In any case, sparse sampling within the compressed sensing framework appears as a valuable option in data-driven computational mechanics applications.

3.3.3 Model order reduction

We consider a last possible application of compressed sensing. It concerns the application to model order reduction, particularly in its hyper-reduction variant, revisited below.

Standard discretization of a given model in the form of a PDE equipped with suitable initial and boundary conditions leads to a linearized system
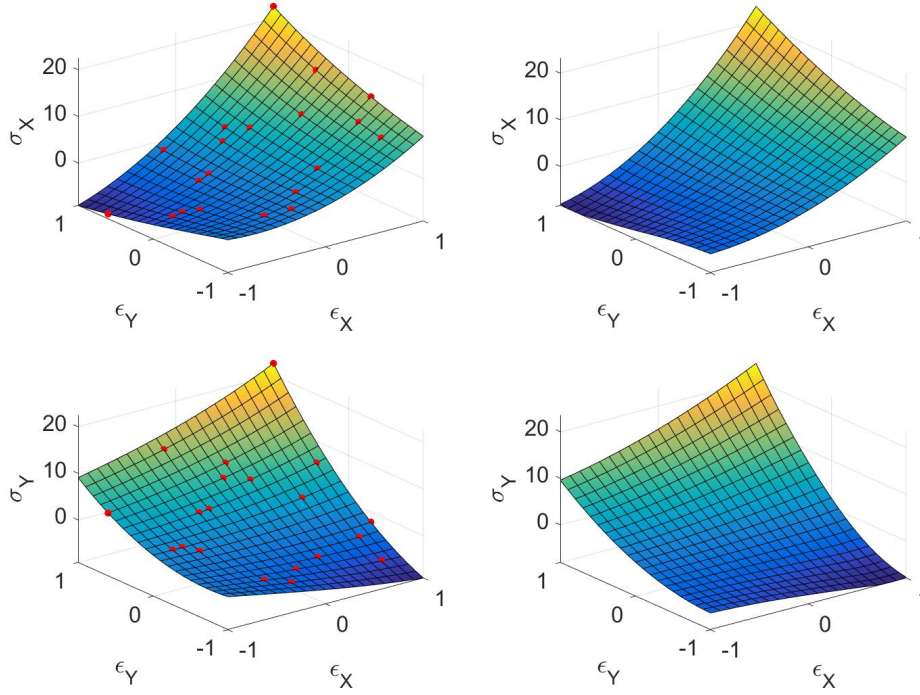
$$\mathbf{KU} = \mathbf{G}, \tag{3.37}$$

Figure 3.53 – Reference (left column) versus compressed sensing based reconstructed behavior manifold (right column). Red points indicate sampling locations.

where, as usual, $\mathbf{K}$ represents the tangent stiffness matrix, $\mathbf{U}$ a vector containing the nodal degrees of freedom, and $\mathbf{G}$ the nodal force vector.

When considering a reduced basis—based on the application of POD or RB methodologies, for instance—, the unknown vector $\mathbf{U}$ can be projected onto the reduced basis according to $\mathbf{U} = \mathbf{Bu}$. The size of vector $\mathbf{u}$ is in general much smaller than the size of the original unknown vector $\mathbf{U}$. Here, $\mathbf{B}$ represents the basis transformation matrix, whose columns are the nodal description of the approximation functions involved in the reduced basis. Thus, the original algebraic system can be rewritten as

$$\mathbf{KBu} = \mathbf{G},$$

that premultiplying by the transpose of $\mathbf{B}$ leads to the reduced system

$$\mathbf{B}^T\mathbf{KBu} = \mathbf{B}^T\mathbf{G} = \mathbf{g},$$

that can be viewed as a Galerkin discretization operating with the reduced basis instead of the one related to the usual finite element approximation.

Within this scenario and assuming sparsity in a target space (e.g. discrete cosines, Fourier, wavelet, ...) for the unknown vectors, $\mathbf{G}$ and $\mathbf{U}$ respectively, allows writing, after assuming the transformation of the unknown vector, expressed from the matrix $\mathbf{T}$,

$$\begin{cases} \mathbf{T}^T\mathbf{G} = \hat{\mathbf{G}} \\ \mathbf{U} = \mathbf{T}\hat{\mathbf{U}} \end{cases},$$

and consequently

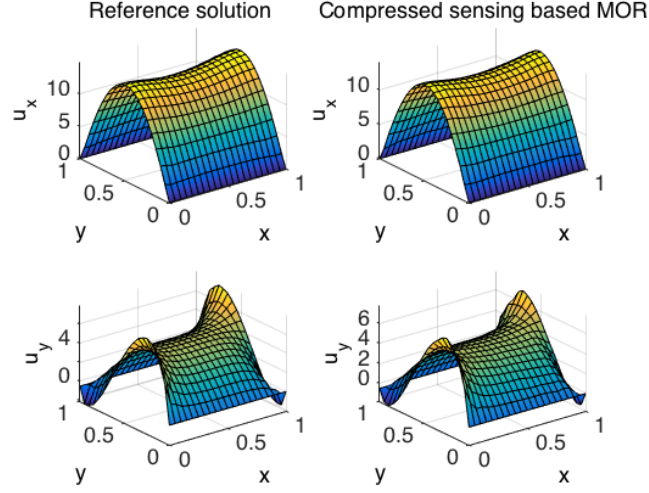$$\mathbf{T}^T\mathbf{KT}\hat{\mathbf{U}} = \hat{\mathbf{G}},$$

Figure 3.54 – Reference versus compressed sensing-based model order reduction.

on which the extraction applies.

To show the potential of this proposal we consider again the discrete system that results from the discretization of a plane-stress elastic problem. Figure 3.54 depicts the reference solution and the one resulting from the reduced formulation just described when extracting randomly 75% of the involved equations, proving the potential of the proposed methodology.

3.3.4 Time-dependent Problems

In this part we will develop a space-time approach for the solution of transient problems with the help of compressed sensing techniques. We will study three different partial differential equations. Namely, steady, transient Poisson and wave equations will be considered, Eqs. (3.38), (3.39) and (3.40), respectively:

$$\alpha \Delta u(\mathbf{x}, t) = b(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega, \tag{3.38}$$

$$\frac{\partial u}{\partial t} - \alpha \Delta u(\mathbf{x}, t) = b(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega, \tag{3.39}$$

and

$$\frac{\partial^2 u}{\partial^2 t} - \alpha \Delta u(\mathbf{x}, t) = b(\mathbf{x}, t) \quad \forall \mathbf{x} \in \Omega. \tag{3.40}$$

These equations must be equipped with suitable Dirichlet boundary conditions at some part of the boundary $\partial \Omega_D$ in order to make the solution unique. For the sake of simplicity, but without loosing generality, we will impose homogeneous Dirichlet boundary conditions at the Dirichlet portion of the Domain $\Omega = [0, 1] \times [0, 1]$, Ω_D,

$$u(x = 0, y) = 0.$$

When dealing with time derivatives, initial conditions have to be imposed,

$$u(\mathbf{x}, 0) = 0,$$

and, possibly, in the case of the wave equation also,

$$\dot{u}(\mathbf{x}, 0) = 0.$$

We will consider a source term that varies in time as,

$$b(\mathbf{x}, t) = A \cos \omega t.$$

To approximately solve Eqs. (3.38), (3.39) and (3.40), finite elements in space and finite differences in time have been used. Therefore, equilibrium for steady/transient Poisson and wave equations at the i-th time step read

$$\mathbf{K}\mathbf{u}^i = \mathbf{f}^i,$$

$$\frac{1}{\Delta t}\mathbf{M}(\mathbf{u}^i - \mathbf{u}^{i-1}) + \mathbf{K}\mathbf{u}^i = \mathbf{f}^i,$$

and

$$\frac{1}{\Delta t^2}\mathbf{M}(\mathbf{u}^{i+1} - 2\mathbf{u}^i + \mathbf{u}^{i-1}) + \mathbf{K}\mathbf{u}^i = \mathbf{f}^i,$$

respectively.

Assume now that we prefer to solve every time step together. We should write a single system of equations of the form

$$\mathbf{A}\hat{\mathbf{u}} = \hat{\mathbf{f}}. \tag{3.41}$$

Here, matrix $\mathbf{A}$ is a block-diagonal matrix containing a matrix $\mathbf{K}$ at each block for the steady Poisson case. The first and second time derivatives appearing in transient Poisson and wave equations will generate coupling terms between consecutive time steps by means of the $\mathbf{M}$ matrices. $\hat{\mathbf{u}}$ and $\hat{\mathbf{f}}$ are the concatenation of spatial nodal unknowns and forces for every time step.

Solving directly the system (3.41) is a legit approach. Indeed, the usual time-marching approach is recovered. However, we would like to explore the advantages of compressed sensing. If the unknown field $\hat{\mathbf{u}}$ is projected onto a new basis enabling a sparse representation, a hyper-reduction results, as seen in the previous section. Therefore, an underdetermined system needs to be solved by performing a L1-norm minimization. Thus, we will seek to solve the following system:

$$\mathbf{E}\mathbf{A}\mathbf{T}\hat{\mathbf{c}} = \mathbf{E}\hat{\mathbf{f}},$$

where $\mathbf{T}$ is again the projection matrix, and $\mathbf{E}$ is the extraction operator that defines randomly which rows are going to be selected to perform the L1 minimization. Finally, $\hat{\mathbf{c}}$ are the unknown coefficients in the new basis. When a time step is chosen, every spatial node related to this time step is automatically selected to keep spatial global equilibrium.

Since we know that the dependance in time of the source term was caused by a cosine function, a smart choice for the projection base in our case is the discrete cosines transform (DCT). It is worth to say that the choice of the projection basis is problem dependent. For instance, if our excitation force evolves as a Heaviside step function, it may be convenient to use a Haar-wavelet-based projection.

3.3.4.1 Numerical results

In this section, several numerical examples involving the three equations considered in the previous section will be analyzed. Sparsity of the new basis will play an important role to make the LASSO algorithm efficient. Parameters employed in the solution of Eqs. (3.38)-(3.40) are compiled in Table 3.3.

The first equation to be tested is Eq. (3.38). The global system of equations is uncoupled from time step to time step due to the fact that it does not involve a time derivative. Fig. 3.55 depicts the value of $u(\mathbf{x} = (0.5, 0.5), t)$ for both an usual time-marching approach and

Parameter	Value
T_f, sim. time	0.1
ω	125.6
Number of time steps	200
α	1
A	-100
Ω	$[0,1] \times [0,1]$
Spatial dofs	16×16

Table 3.3 – Parameters employed in the analysis of Eqs. (3.38)-(3.40).

for a compressed sensing-based procedure. In this case only 5 time instants out of 200 are considered in the LASSO minimization. As it can be noticed, the blue curve tends to capture the overall behavior of the red curve (reference solution). However, there are still some noisy peaks appearing in the blue curve due to the small number of time instants involved in the computation. Fig. 3.56 shows the coefficients of the discrete cosines transform, which is indeed what the LASSO algorithm computes. It can be noticed how some high frequency peaks are appearing causing the small oscillations in the u field.

The results are better if we increase the number of sampled time instants up to ten. Figs. 3.57-3.58 show u and c fields, respectively. It should be highlighted that the high-frequency peaks are no longer appearing. Furthermore, the leakage pollution close to the main peak is filtered thanks to the LASSO algorithm.

The case of the wave equation (3.40), with a second derivative in time, is less problematic than the one involving first derivatives only, see Eq. (3.39), since it forces the response to follow the loading, whereas the transient Poisson equation involves diffusion. Therefore, we will expect a sparser solution than in problem (3.39).

Fig. 3.59 shows the u field for the wave equation case. Since the source term is a cosine function, the response of the system is also a cosine. Furthermore, the system must be initialized with a non-homogeneous initial condition. Otherwise, compatibility conditions of the time marching approach will no longer be satisfied. Fig. 3.60 shows the DCT coefficients $\mathbf{c}$ for the wave equation case. As it can be noticed, the first coefficient accounts for a solid rigid-like translation (i.e., the cosine is not centered in 0) and the other peak coincides with the frequency of the source term.

Figs. 3.61, 3.62, 3.63, and 3.64 show the expected behavior: imposing a cosine in the source term with a first time derivative will cause a sinusoidal response, which is not sparse in the projected base, due to the diffusion effects. Hence, more time instants need to be taken into account in order to achieve an accurate result.

3.3.5   Conclusions

In this work we explored different applications of compressed sensing in computational mechanics. First, we proved that it could be a valuable strategy for performing random samplings to evaluate solutions of parametric models. Then, inspired from the so-called "single pixel camera", we analyzed the use of the compressed sensing methodology to reconstruct constitutive manifolds from the only knowledge of a quite reduced number of data-points. Finally, , we proved that the same methodology can be employed for reducing
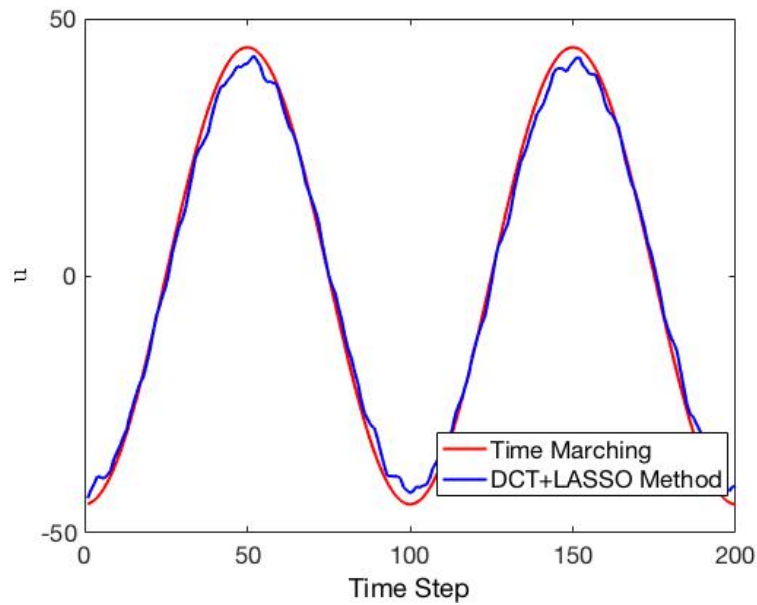
Figure 3.55 – $u(\mathbf{x} = (0.5, 0.50), t)$ for steady Poisson's case. Five time instants are sampled out of two hundred possible time steps.
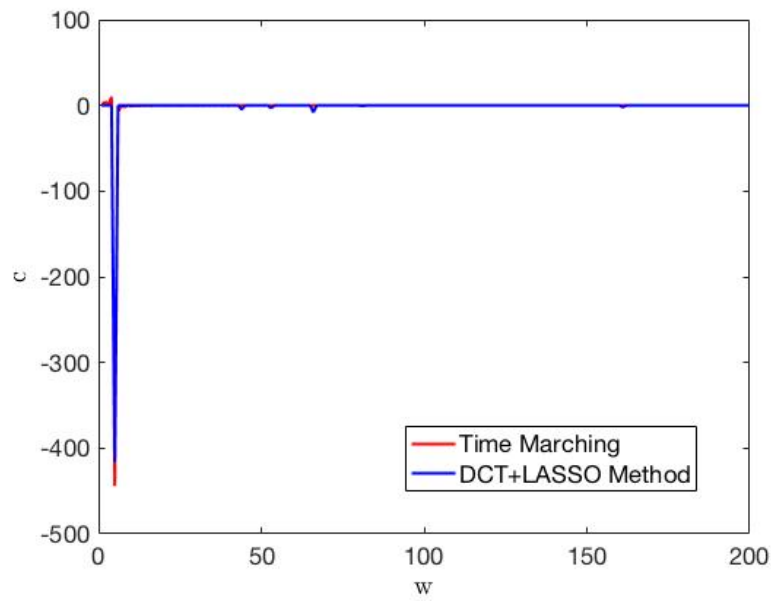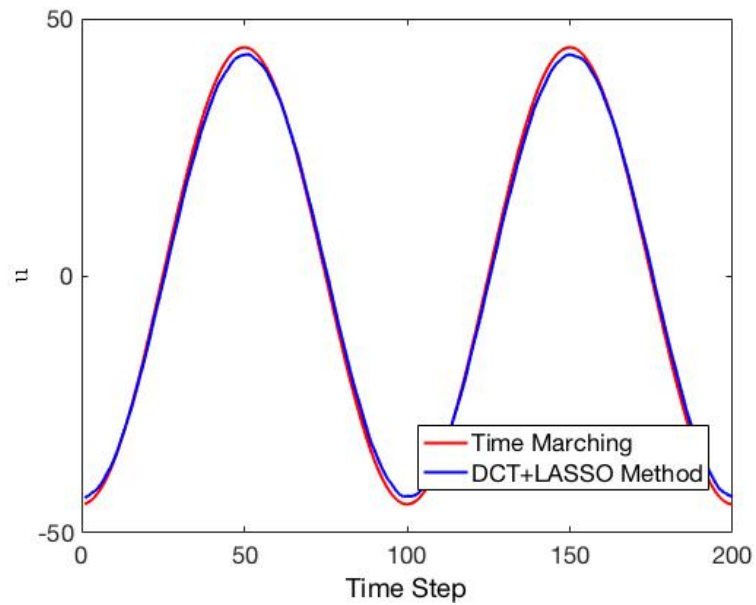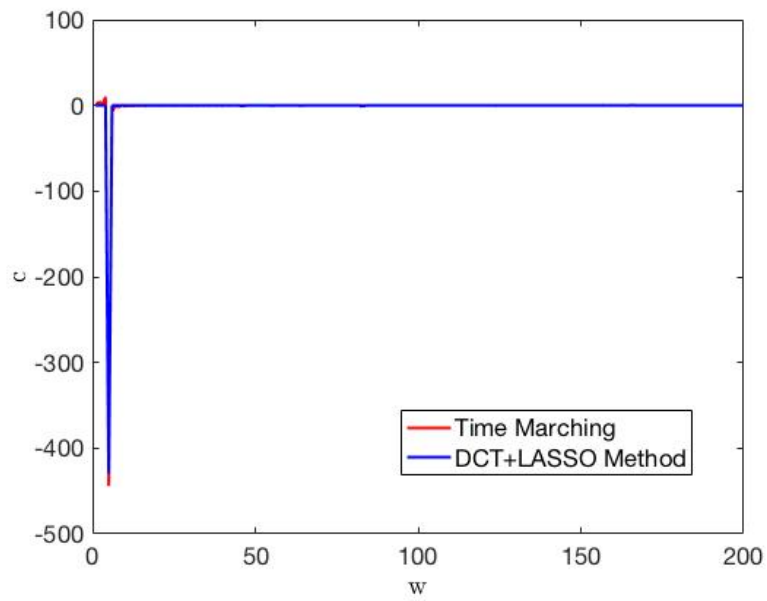

Figure 3.56 – $c(\mathbf{x} = (0.5, 0.50), \omega)$ for steady Poisson's case. Five time instants are sampled out of two hundred possible time steps.
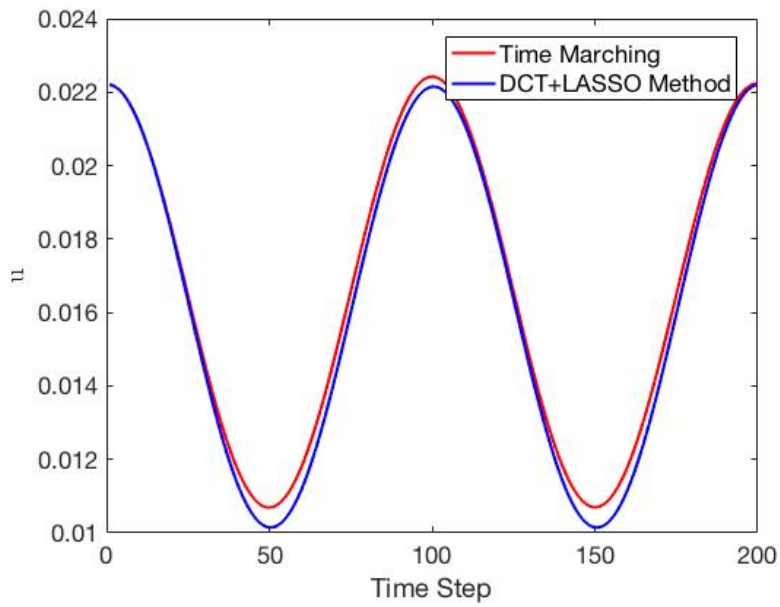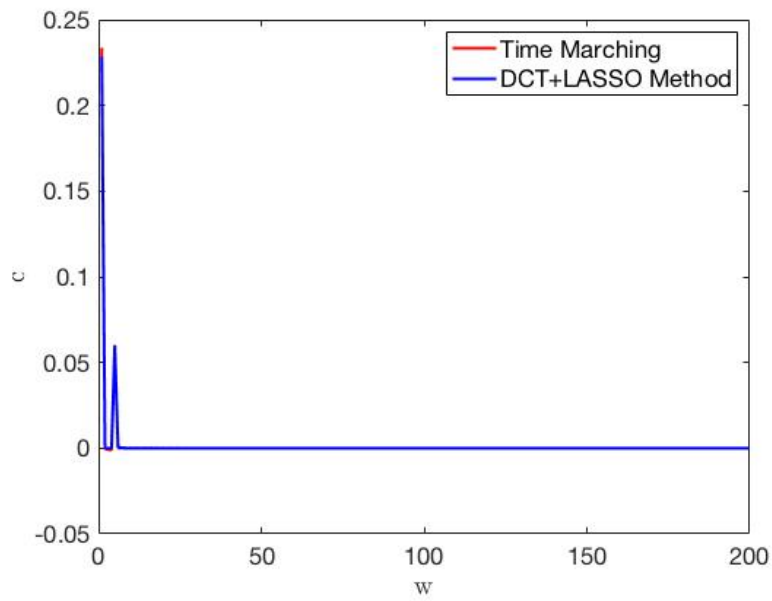
Figure 3.57 – $u(\mathbf{x} = (0.5, 0.50), t)$ for steady Poisson's case. Five time instants are sampled out of two hundred possible time steps.

Figure 3.58 – $c(\mathbf{x} = (0.5, 0.50), \omega)$ for steady Poisson's case. Five time instants are sampled out of two hundred possible time steps..

Figure 3.59 – $u(\mathbf{x} = (0.5, 0.50), t)$ for the wave equation. 30 time instants are sampled out of two hundred possible time steps.
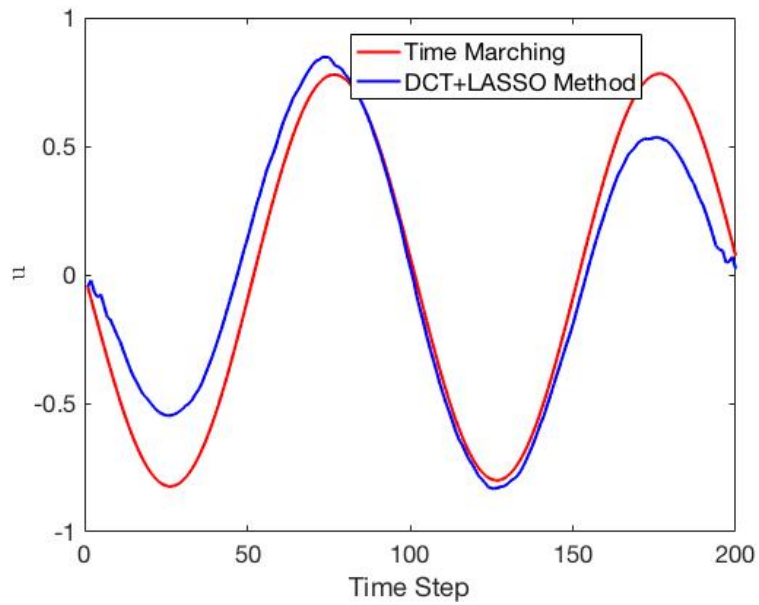

Figure 3.60 – $c(\mathbf{x} = (0.5, 0.50), t)$ for the wave equation. 30 time instants are sampled out of two hundred possible time steps.

Figure 3.61 – $u(\mathbf{x} = (0.5, 0.50), t)$ for transient Poisson's case. 100 time instants are sampled out of two hundred possible time steps.
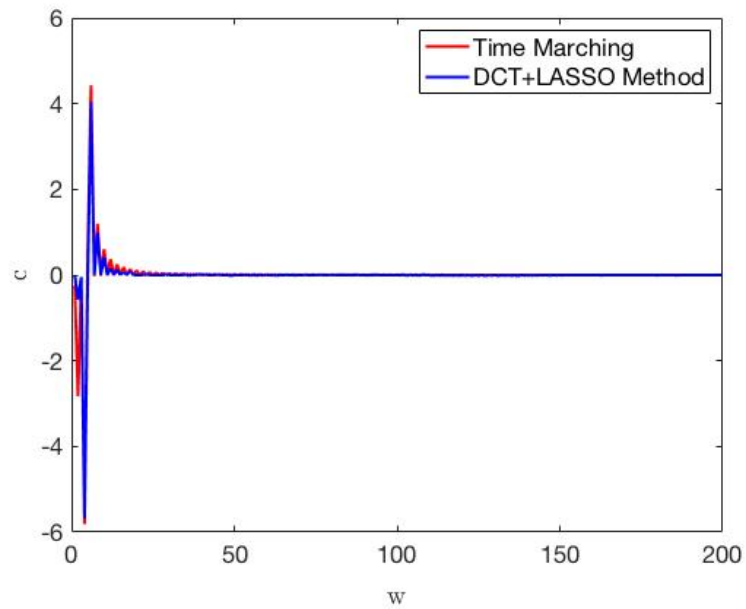

Figure 3.62 – $c(\mathbf{x} = (0.5, 0.50), t)$ for transient Poisson's case. 100 time instants are sampled out of two hundred possible time steps.
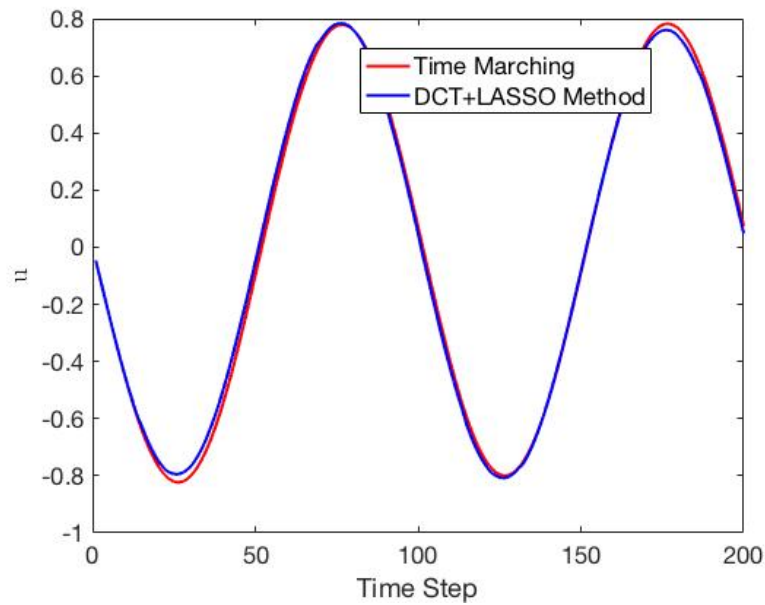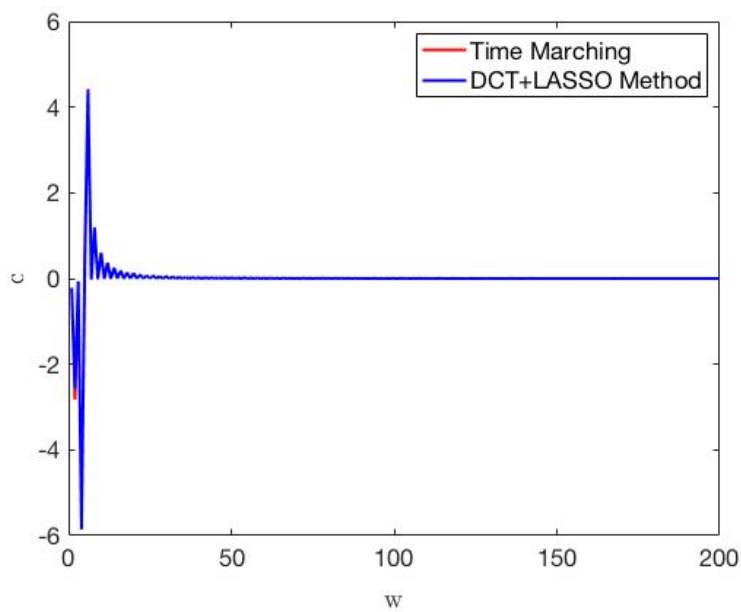
Figure 3.63 – $u(\mathbf{x} = (0.5, 0.50), t)$ for transient Poisson's case. 150 time instants are sampled out of two hundred possible time steps.

Figure 3.64 – $c(\mathbf{x} = (0.5, 0.50), t)$ for transient Poisson's case. 150 time instants are sampled out of two hundred possible time steps.

significantly the size of the discrete system of equations that results from the application of standard discretization techniques defining a new kind of model order reduction techniques.

All these different application rely on a single technique and we strongly believe that it will play a very relevant role in computational mechanics for the years to come.

Part II

Advances on Model Order Reduction Techniques

Even though Dynamic Data Driven Application opens a new paradigm in the computational mechanic framework, big steps has to be considered before transforming it into a tangible reality. Appart from all difficulties and advantages arising when working with data, it is equally important to keep in mind that these data corrections have to be performed in a dynamic way, or in other words, as fast as possible. Model Order Reduction Techniques is the perfect ingredient to accomplish such constraints. Plenty of different MOR techniques have been proposed in the recent years to alleviate computational costs. The purpose of this chapter is to present some advances on the field of Model Order Reduction Techniques since we strongly believe that those advances will play a fundamental role within the DDDAS framework.

This part is divided into two different chapters. Chapter 4 proposes a methodology that uses multiple PGDs to describe a given solution. A standard macro FEM partition of the domain is in charge of controlling the overlap between different PGDs. A greater separability of the solution is achieved, due to the fact that each PGD will be devoted to a local portion of the domain, being the solution more linear compared to the global one.

Chapter 5 explores new applications of the so-called PGD in-plane-out-of-plane separated representation. This kind of representation becomes particularly useful when dealing with degenerated domains in which the out-of-plane dimension is much smaller than the other two dimensions contained in the plane coordinates.

Enhancing PGD Separability based on Partition of Unity Method

Abstract   Model order reduction techniques become very useful within the Dynamic Data Driven Application System framework where decision making has to be done in a dynamic way. Indeed, strategies where parametric solutions are precomputed off-line will allow to make faster such decision making. However, it is well known that model order reduction techniques that project the solution of the problem at hand onto a low-dimensional subspace present difficulties when this solution lies on a non-linear manifold. To overcome these difficulties—notably, an undesirable augment in the number of required modes in the solution—several solutions have been suggested. Among them we can cite the use of non-linear dimensionality reduction techniques or, alternatively, the employ of local linear reduced order approaches. These last approaches usually present the difficulty of ensuring continuity between these local models. In this chapter, a new method is presented that ensures this continuity by resorting to the paradigm of the partition of unity (PU), while employing Proper Generalized Decompositions at each local patch.

Contents

Despite of the fact that PGD has been proven to be very effective generating parametric solutions in many situations [Mena *et al.* 2015, Chinesta *et al.* 2013a], its performance is decimated when dealing with highly non-separable solutions i.e. plenty of modes are required to obtain an accurate solution. This chapter explores the possibility of enhancing

the perfomance of standard PGD algorithm by means of merging the Partiton of Unity (PU) ideology [Babuska & Melenk 1996] [Babuska & Melenk 1997] together with the PGD rationale. Basicly, a coarse partition of the domain guarantees a smooth transition between the different PGDs coexisting in the domain. Section 4.1 presents the main ideas behind the PU-PGD rationale, the algorithm is tested in several partial differential equations ranging from pure diffusion to transient equations. Section 4.2 extends the PU-PGD in order to handle multiscale problems. A continuous transition between scales is automaticly inherited by virtue of the consistent PU method.

These topics are addressed in two submitted papers:

- R. Ibáñez, E. Abisset-Chavanne, F. Chinesta, A. Huerta, E. Cueto, *A local, multiple Proper Generalized Decomposition based on the Partition of Unity*. Submitted to International Journal Numerical Methods for engineering, 2018.

- R. Ibáñez, A. Ammar, E. Cueto, A. Huerta, J.-L. Duval, F. Chinesta, *Multi Scale Proper Generalized Decomposition based on the Partition of Unity*. Submitted to International Journal Numerical Methods for engineering, 2018.

4.1 A local, multiple Proper Generalized Decomposition based on the Partition of Unity

The main objective of this section is to develop a generalized multi PGD formulation in which continuity between subdomains is guaranteed. This will be achieved by resorting to the Partition of Unity (PU) paradigm [Babuska & Melenk 1996] [Babuska & Melenk 1997]. By employing the partition of unity, continuity of the solution is guaranteed if the chosen PU is continuous. This will allow us to glue different PGD approximations defined at particular regions of the space, time or parameter spaces, thus ensuring a global solution with a minimum of degrees of freedom.

4.1.1 Methodology

A method based on combining PU/PGD approaches is proposed in this section. On one hand FE shape functions are widely used in the computational mechanics field since they exhibit excellent properties as local behaviour or partition of unity. On the other hand, the PGD or low rank approximation approaches are very useful when dealing with domains where separation of variables is possible. Therefore, we would like to combine both approaches to achieve both geometrical adaptivity and high resolution constraints.

For the sake of simplicity but without loosing generality, the method is ilustrated with a weak form related to a general 2D PDE constraint to the domain Ω as shown in eq. (4.1). Where the right hand side $\mathcal{F}(\cdot, \cdot)$ takes into account both the external forces and the appropiate set of boundary conditions to ensure the well-posedness of the problem.

$$\mathcal{L}(u^*(x,y), u(x,y)) = \mathcal{F}(u^*(x,y), f(x,y)) \in \Omega \tag{4.1}$$

Normally, solving the weak form (4.1) requires an approximation space for the primal variable $u(x,y)$ and its associated variation $u^*(x,y)$. Eq. (4.2) depicts classical finite element approximation, where $\mathcal{I}$ is the entire set of shape functions defined over the integration domain, Ω.

$$u(x,y) = \sum_{i \in \mathcal{I}} N_i(x,y) u_i \tag{4.2}$$

Furthermore, standard FE shape functions satisfy the partition of unity and both linear compatibility properties as shown in (4.3), respectively.

$$\sum_{i \in \mathcal{I}} N_i(x,y) = 1 \qquad \forall \ x, y \in \Omega$$

$$\sum_{i \in \mathcal{I}} N_i(x,y) x_i = x \qquad \forall \ x, y \in \Omega$$

$$\sum_{i \in \mathcal{I}} N_i(x,y) y_i = y \qquad \forall \ x, y \in \Omega \qquad (4.3)$$

As mention before, we would like to combine both shape functions coming from FE ($N_i(x,y)$) and low rank approximation methods (PGD). Therefore, each degree of freedom u_i associated to the FEM shape function $N_i(x,y)$ will be replaced with a PGD functional approximation, aiming to capture the details of the solution which are not captured by the standard FEM coarse meshes. Therefore, the approximation of the solution will read as shown in Eq. (4.4).

$$u(x,y) = \sum_{i \in \mathcal{I}} N_i(x,y) \sum_{k=1}^{M} X_k^i(x) Y_k^i(y) \qquad (4.4)$$

Where $X_k^i(x)$ and $Y_k^i(x)$ functions are the k-th unidimensional modes related to the i-th PGD.

Several approaches can be adopted to define the trial function. In this particular case, a Galerkin projection is selected. Hence, the same approximating space is choosen for $u(x,y)$ and $u^*(x,y)$ as shown in eq. (4.5).

$$u^*(x,y) = \sum_{i \in \mathcal{I}} N_i(x,y)(X_M^{i*}(x) Y_M^i(y) + X_M^i(x) Y_M^{i*}(y)) \qquad (4.5)$$

It can be highlighted the Greedy nature of the algorithm since the variation only takes into account the last M-th PGD mode taking the previous modes as known. Furthermore, a non-linear system, which has been created due to the separation of variables, is solved using an alternate direction scheme. Indeed, $Y_M^{i*}(y)$ is set to zero when looking for modes in the x direction and analogously $X_M^{i*}(x)$ vanishes when looking for modes in the y direction.

The properties of the method will change depending on the way finite element shape functions are defined. Fig. (4.1) exemplifies the partition when piecewise constant shape functions are used. As it can be seen, no overlapping between different PGDs exists. Therefore, extra effort has to be done to set proper interface conditions along the red lines appearing in the same figure.

Fig. (4.2) depicts a piecewise linear partition of the domain. As it can be seen, an element has contributions coming from four different PGDs. The coupling conditions between different PGDs are automaticly taken into account in the elemental contributions. Indeed, this kind of partition imposes a smooth transition between PGDs which varies continuously along the domain in accordance with the global partition of the domain.

It is important to mention that the full potential of PGD approximation resides in the capability of writing the integral form in a separated manner. By doing that, all integrals in a high dimensional space will be splitted into a product of integrals related to each one of the subspaces. For the sake of simplicity, we will assume that the shape functions $N_i(x,y)$ are separable in one mode as shown in eq. (4.6). It is important to notice that it will be the case for instance of 2D quadrilateral elements with straight and paralel sides.
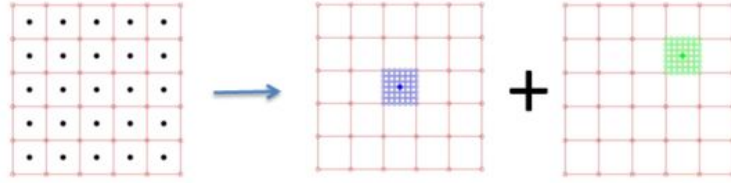
Figure 4.1 – Left, constant piecewise partition. Center, domain of influence of i-th PGD. Right, domain of influence of k-th PGD.
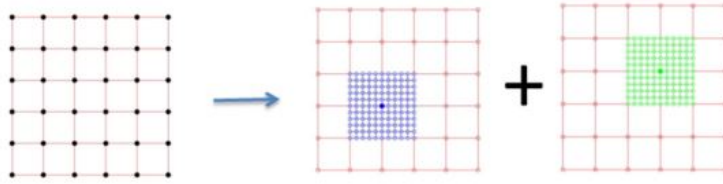

Figure 4.2 – Left, linear piecewise partition. Center, domain of influence of i-th PGD. Right, domain of influence of k-th PGD.

$$N_i(x,y) = N_i^x(x)N_i^y(y) \tag{4.6}$$

Even though this assumption may be seen as a limitation, it is important to reckon that there are some variables such as the time or parametres that normally admits cartesian decompositions since they vary in a closed interval. Therefore, when dealing with a complex spatial geometry that evolves in time, a smart partition would be to keep the space variable as it is, but making a high order partition in time, generating a prism like FE shape function. Nevertheless, this very first work is meant to provide an insight of the method, thus, all results presented in the sequel will be related to piecewise linear FE shape functions based on squared elements.

4.1.2 m-PGD in approximation

In this section, we proposed to analyze the capability of the proposed methodology in eq. (4.7). As it can be seen, there are no partial derivatives involved, thus, just a compact expression of the function $f(x,y)$ is seeked.

$$\int_\Omega u^*(x,y)u(x,y)dxdy = \int_\Omega u^*(x,y)f(x,y)dxdy \tag{4.7}$$

Following the spirit of standard finite element approximation the integral over the entire domain Ω is splitted into a sum of integrals for each one of the finite elements Ω_e appearing in the domain. Just as a recall, piecewise linear shape functions present contributions from four different PGDs per element (one for each corner), as shown in Fig. (4.3).

It is important to notice that the i-th PGD is defined in four different elements (except from boundary PGDs), thus, special attention has to be paid when doing the assembly of the global matrix. Taking into account the numbering appearing in fig. (4.3), second and
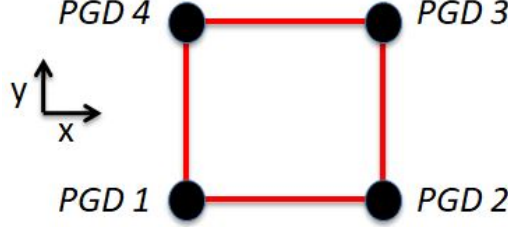
Figure 4.3 – PGDs distribution for piecewise linear quad elements.

third local PGDs are assembled in the lower half part of the x modes, whereas first and fourth local PGDs are assembled in the higher half part of the y modes. Analogously, third and fourth PGDs are assembled in the lower half part of the y mode, whereas first and second local PGDs are assembled in the higher half part of the y mode.

Omitting the dependance with respect to space variables x, y and introducing the separability of the FE shape function 4.6, the left hand side integral form of (4.7) particularized for a element Ω_e reads as shown in eq. (4.8).

$$\int_{\Omega_e} u^* u \, dx dy = \sum_{i=1}^{4} \sum_{j=1}^{4} \int_{\Omega_e} N_i^x N_i^y (X_M^i Y_M^i)^* (N_j^x N_i^y \sum_{k=1}^{M} X_k^j Y_k^j) dx dy \tag{4.8}$$

The next step is to split the 2D integral form into a product of 1D integrals as shown in eqs. (4.9-4.10). Where variations for x and y systems are presented, respectively. As it can be seen, the variation of the x direction shares all the operators with the variation of the y direction, it is just a matter of where the variation is placed. Indeed, this properties comes naturally from the alternate direction scheme that is used to solve the non-linear system of the PGD.

$$\int_{\Omega_e} u^* u \, dx dy = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{M} \int_x X_M^{i*} N_i^x N_j^x X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy \tag{4.9}$$

$$\int_{\Omega_e} u^* u \, dx dy = \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{M} \int_x X_M^i N_i^x N_j^x X_k^j dx \int_y Y_M^{i*} N_i^y N_j^y Y_k^j dy \tag{4.10}$$

Indeed, the efficiency of the m-PGD algorithm is maximized when all operators related to eqs. (4.9-4.10) are precomputed in an off-line phase as shown in eqs. (4.11-4.12).

$$\alpha_x^{ijk} = \int_x X_M^i N_i^x N_j^x X_k^j dx \tag{4.11}$$

$$\alpha_y^{ijk} = \int_y Y_M^i N_i^y N_j^y Y_k^j dx \tag{4.12}$$

Analogously, the right hand side term appearing in eq. (4.7) reads as shown in eqs. (4.13-4.14), for x and y systems respectively.

$$\int_{\Omega_e} u^* f \, dx dy = \sum_{i=1}^{4} \sum_{z=1}^{Z} \int_x X_M^{i*} N_i^x f_z^x dx \int_y Y_M^i N_i^y f_z^y dy \tag{4.13}$$
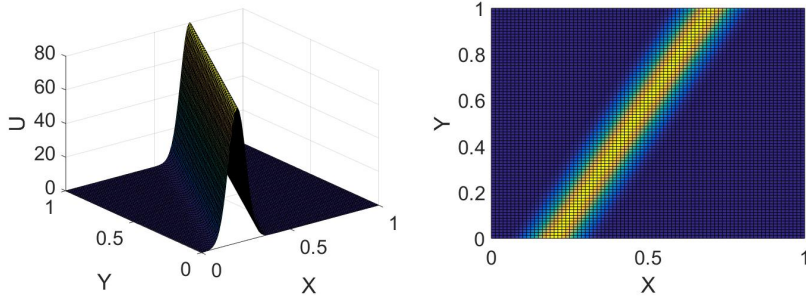
Figure 4.4 – Left, 3D view of $f_1(x,y)$. Right, planar view of $f_1(x,y)$.

$$\int_{\Omega_e} u^* f dx dy = \sum_{i=1}^{4} \sum_{z=1}^{Z} \int_x X_M^i N_i^x f_z^x dx \int_y Y_M^{i*} N_i^y f_z^y dy \qquad (4.14)$$

Where the source term f(x,y) is expressed in a separated format as shown in eq. (4.15).

$$f(x,y) = \sum_{z=1}^{Z} f_z^x(x) f_z^y(y) \qquad (4.15)$$

Once again, all operators related to the integration of the source term can be precomputed off-line as shown in eqs. (4.16-4.17).

$$\varepsilon_x^{iz} = \int_x X_M^i N_i^x f_z^x dx \qquad (4.16)$$

$$\varepsilon_y^{iz} = \int_y Y_M^i N_i^y f_z^y dy \qquad (4.17)$$

Let's ilustrate the methodology using a highly non separable function given in eq. (4.18). Where $\sigma = 0.05$, $v = 0.5$ and $x_0 = 0.2$.

$$f_1(x,y) = \frac{10}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(vy+x_0))^2}{2\sigma^2}} \qquad (4.18)$$

Fig. (4.4) depicts the resulting $u(x,y) = f_1(x,y)$ scalar field from different perspectives. As it can be seen, the source term is going through the antidiagonal of the domain, generating a highly non-separable function. Indeed, a normal PGD algorithm suffers to capture such kind of solution.

Two different partitions of the domain has been tested as shown in fig. (4.5). PGDs acting on left and right sides are set to zero since the solution is equal to zero in these two regions. Nevertheless, PGDs acting on bottom and top parts are active since the solution is different from zero there. The red points indicate the centroid location for all PGDs in the domain. The elements are coloured in accordance with the number of active PGDs acting in each element.

Fig. (4.6) shows the reconstructed solution using 4 modes per local PGD, when the domain has 8 active PGDs (left) and 24 PGDs (right). As it can be seen, even though both approximations capture the main features of the solution, the solution using 24 PGDs is slightly better than the one with 8 PGDs.
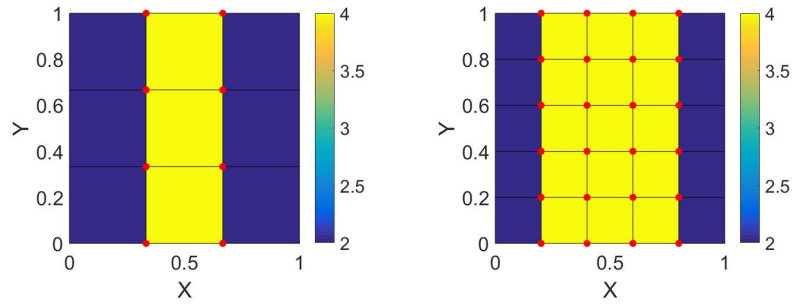
Figure 4.5 – Left, domain partioned into 8 PGDs. Right, domain partitioned into 24 PGDs. Red points, centroid of each PGD. Element colour, number of PGDs per element.
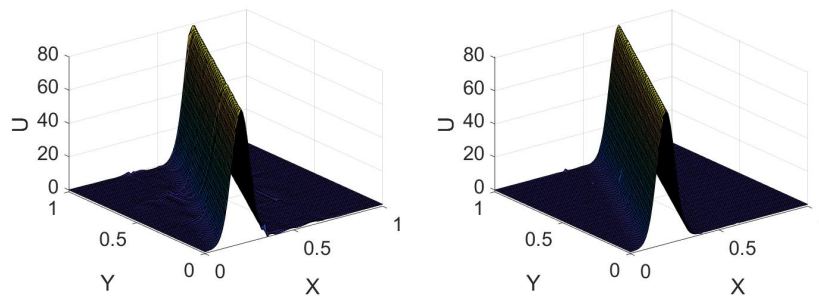

Figure 4.6 – Left, solution with 8 PGDs. Right, solution with 24 PGDs. Both solutions have four modes.
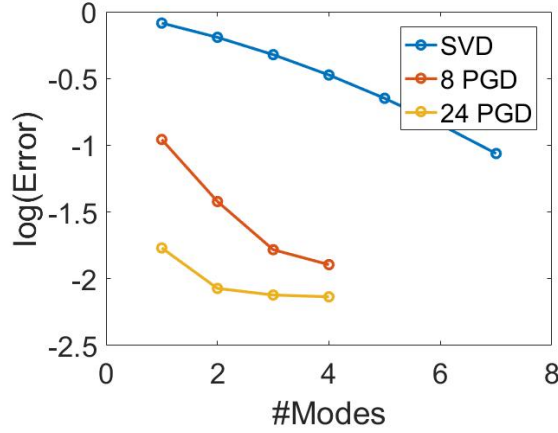
Figure 4.7 – Convergence error for approximation problem in logarithmic scale for different number of modes. Comparison between SVD and two different partitions of the domain.

Fig. (4.7) shows the convergence error for the approximation of $f_1(x, y)$. The relative error is measured as shown in eq. (4.19), where u^m is the reconstructed solution using m modes. As it can be seen, singular value decomposition (SVD) method suffers from separating this kind of solutions, having a slow decay of the relative error with respect to the number of modes. On the other hand, using the multi PGD algorithm the relative error decays much faster than the one related to the SVD.

$$\mathcal{E}^m = \frac{||u_{ref} - u^m||}{||u_{ref}||} \tag{4.19}$$

It is important to keep in mind that the more a domain is partitioned the faster the relative error decays with the number of modes per element. This behaviour is expected since the smaller the elements are, the easier is to capture a local behaviour. However, the price to pay is that the cost of computing one mode per PGD will increase just like the storaging cost as well.

4.1.3 m-PGD for different PDEs

The aim of this section is to show the potential of the proposed methodology when it is applied to different PDEs. The complexity of the PDEs is going to increase from test case to test case. Therefore, the first test case is a fully diffusive equation. Then, the second case is a convection-reaction-diffusion equation. Finally, the last example relates to a transient heat thermal problem with a moving source term. Even though the construction of the operators for each PDE follows the same trend than the ones presented in the m-PGD in approximation, further details are given in the annex.

4.1.3.1 Diffusion PDE

The first PDE to be tested is the one shown in eq. (4.20). Null Dirichlet boundary conditions are imposed in the entire boundary. The source term is equal to $f_1(x, y)$ and the viscosity parameter (η) is set to 1.
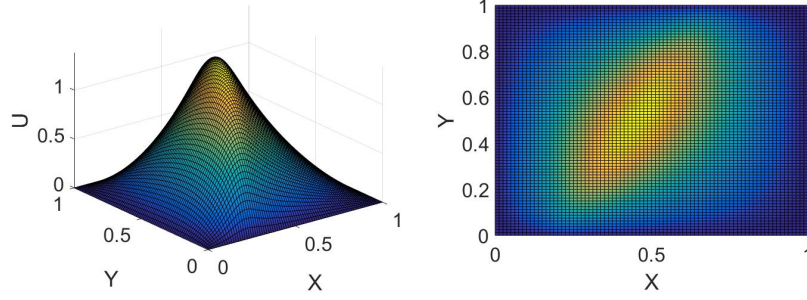
Figure 4.8 – Left, 3D view of $u(x, y)$. Right, planar view of $u(x, y)$ for diffusive equation.

$$\int_\Omega \eta \nabla u^* \cdot \nabla u \, dx dy = \int_\Omega u^* f_1 \, dx dy \tag{4.20}$$

Eq. 4.21 shows the derivation of a diffusive operator in a separated format. For the sake of simplicity but without loosing generality, we will also assume that a new mode in the x direction is desired. Therefore, all variations related to the y direction are set to zero.

$$\int_{\Omega_e} \frac{\partial u^*}{\partial x} \frac{\partial u}{\partial x} dx dy =$$

$$\sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_{\Omega_e} \left(\frac{\partial N_i}{\partial x} X_M^{i*} Y_M^i + N_i \frac{\partial X_M^{i*}}{\partial x} Y_M^i \right) \left(\frac{\partial N_j}{\partial x} X_k^j Y_k^j + N_j \frac{\partial X_k^j}{\partial x} Y_k^j \right) dx dy =$$

$$= \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} \frac{\partial N_i^x}{\partial x} \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy +$$

$$+ \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x \frac{\partial X_M^{i*}}{\partial x} N_i^x \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy +$$

$$+ \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x \frac{\partial X_M^{i*}}{\partial x} N_i^x N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy +$$

$$+ \sum_{i=1}^4 \sum_{j=1}^4 \sum_{k=1}^M \int_x X_M^{i*} \frac{\partial N_i^x}{\partial x} N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy \tag{4.21}$$

Fig. (4.8) depicts the resulting $u(x, y)$ scalar field for the diffusion equation from different perspectives. As it can be seen, the solution diffuses the source term which is going through the antidiagonal of the domain. As it can be seen, the scalar field $u(x, y)$ is set to zero in the entire boundary.

Two different partitions of the domain has been tested as shown in fig. (4.9). PGDs acting on the boundaries are set to zero since null Dirichlet boundary conditions are imposed in the boundary. The red points indicate the centroid location for all PGDs in the domain. The elements are coloured in accordance with the number of active PGDs acting in each element. As it can be seen, elements in the corner have only one active PGD, whereas elements in the sides have two active PGDs and interior elements have four active PGDs.
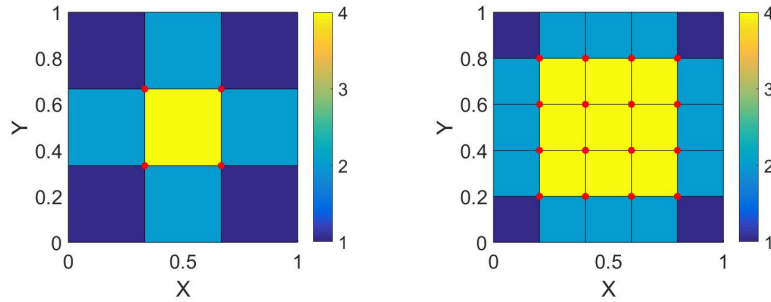
Figure 4.9 – Left, domain partioned into 4 PGDs. Right, domain partitioned into 16 PGDs. Red points, centroid of each PGD. Element colour, number of PGDs per element.
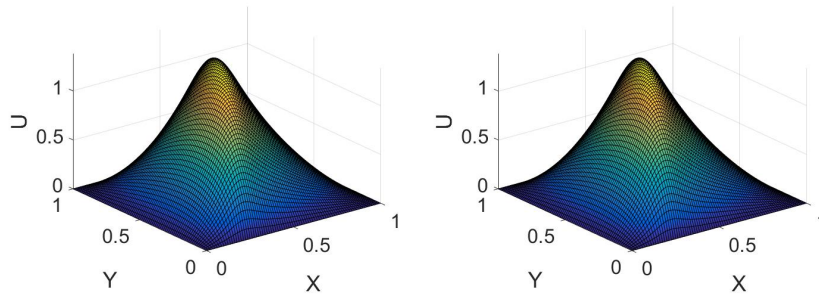

Figure 4.10 – Left, 4 PGDs. Right, 16 PGDs. Both solutions have four modes. Solution of the diffusion equation.

Fig. (4.10) shows the reconstructed solution using 4 modes per local PGD, when the domain has 4 active PGDs (left) and 16 PGDs (right). As it can be seen, both approximations are in perfect accordance with the reference solution.

Fig. (4.11) shows the convergence error for the diffusive equation. As it can be seen, all methods converge monotonicly, being the SVD the one that presents the slower convergence with respect the number of modes. Indeed, using the multi PGD algorithm presents a faster decay in the relative error.

4.1.3.2   Convection-Reaction-Diffusion (CRD) PDE

In this subsection a PDE consisting of a combination of a diffusive, plus a reaction term, plus a convective term is studied. The two first contributions were already studied in the previous examples. Indeed, the major novelty is the introduction of the convective term since it will involve a non-symmetric operator. The weak form related to the CRD PDE is shown in eq. (4.22).

$$\int_\Omega u^* \mathbf{v} \cdot \nabla u + \eta \nabla u^* \cdot \nabla u + \sigma u^* u \; dxdy = \int_\Omega u^* f_2 dxdy \tag{4.22}$$
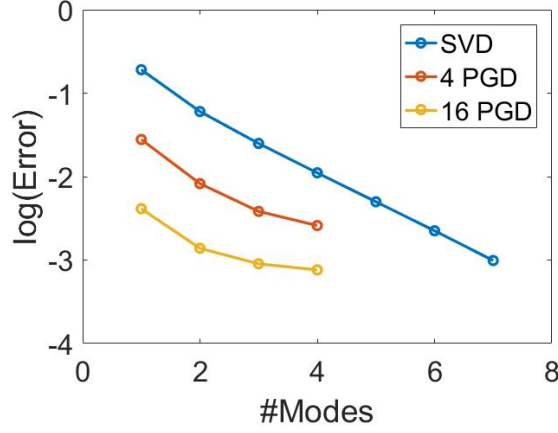
Figure 4.11 – Convergence error for diffusion problem in logarithmic scale for different number of modes. Comparison between SVD and two different partitions of the domain.

Where null Dirichlet boundary conditions are acting on the entire boundary, the integration domain is $\Omega = [0,1] \times [0,1]$, the velocity field is $\mathbf{v} = 500(y - 0.5, 0.5 - x)^T$, the reaction term is set to $\sigma = 10$ and the diffusive term to $\eta = 1$. Regarding the source term, a Gaussian placed in $[x_0, y_0] = [0.75, 0.75]$ as shown in eq. (4.23) is imposed.

$$f_2(x,y) = \frac{800}{\sqrt{2\pi}} e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{0.005}} \tag{4.23}$$

Eq. 4.24 details how to derive a pure convection of a scalar field $u(x,y)$ along the x direction in a separated format. The derivation for the convection along the y direction is analogous. It is imporant to notice that this case is a particular case where the velocity field is given by $\mathbf{v} = (1,0)^T$. If a more complex convective field is considered, a separated representation of the velocity field would be required. For the sake of simplicity but without loosing generality, we will also assume that a new mode in the x direction is desired. Therefore, all variations related to the y direction are set to zero.

$$\int_{\Omega_e} u^* \frac{\partial u}{\partial x} dxdy =$$

$$\sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{M} \int_{\Omega_e} N_i X_M^{i*} Y_M^i \left(\frac{\partial N_j}{\partial x} X_k^j Y_k^j + N_j \frac{\partial X_k^j}{\partial x} Y_k^j \right) dxdy =$$

$$= \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{M} \int_x X_M^{i*} N_i^x \frac{\partial N_j^x}{\partial x} X_k^j dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy +$$

$$+ \sum_{i=1}^{4} \sum_{j=1}^{4} \sum_{k=1}^{M} \int_x X_M^{i*} N_i^x N_j^x \frac{\partial X_k^j}{\partial x} dx \int_y Y_M^i N_i^y N_j^y Y_k^j dy \tag{4.24}$$

Fig. (4.12) depicts the resulting $u(x,y)$ scalar field for the CRD equation from different perspectives. As it can be seen, the gaussian source term is convected circularly according to the velocity convection field. Moreover, the diffusion term makes that the Gaussian is
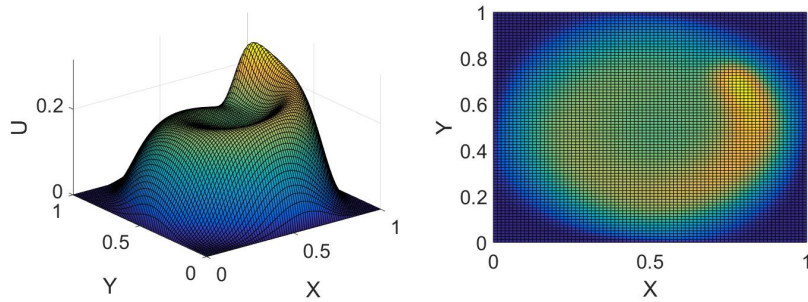
Figure 4.12 – Left, 3D view of $u(x, y)$. Right, planar view of $u(x, y)$ for convection-reaction-diffusion equation.
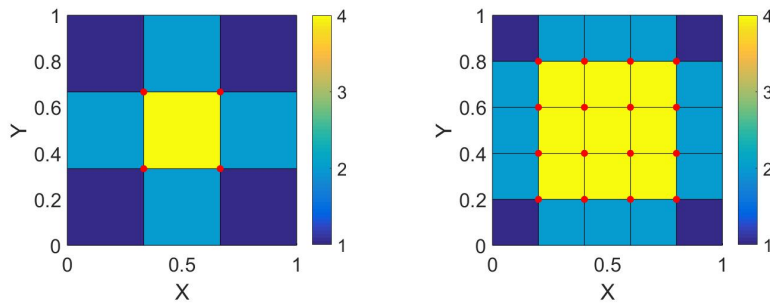

Figure 4.13 – Left, domain partioned into 4 PGDs. Right, domain partitioned into 16 PGDs. Red points, centroid of each PGD. Element colour, number of PGDs per element.

smoothed throughout its convection. It can be also noticed that the scalar field $u(x, y)$ is set to zero in the entire boundary of the domain.

The same partitions than in the diffusive case have been tested as shown in fig. (4.13). PGDs acting on the boundaries are set to zero since null Dirichlet boundary conditions are imposed in the boundary.

Fig. (4.14) shows the reconstructed solution using 4 modes per local PGD, when the domain has 4 active PGDs (left) and 16 PGDs (right). As it can be seen, the approximation related to 4 active PGD with 4 PGD modes does not capture the right solution. Indeed, a higher number of modes will be needed to converge to the reference solution. On the other hand, the partition involving 16 PGDs is in perfect accordance with the reference solution, capturing all the features of the scalar field without presenting any oscillation.

Fig. (4.15) shows the convergence of the relative error in logarithmic scale for the SVD and the two different partitions proposed for the CDR equation. It can be clearly seen that the convergence related to the 4 PGD partition is slowest one. Indeed, this problematic is derived from the fact that the first PGD modes have to capture a highly non-linear behaviour, meaning that this partition will require a high amount of modes to converge to the real solution. On the other hand, the 16 PGD partition convergence is quite fast compared with the other two methods.
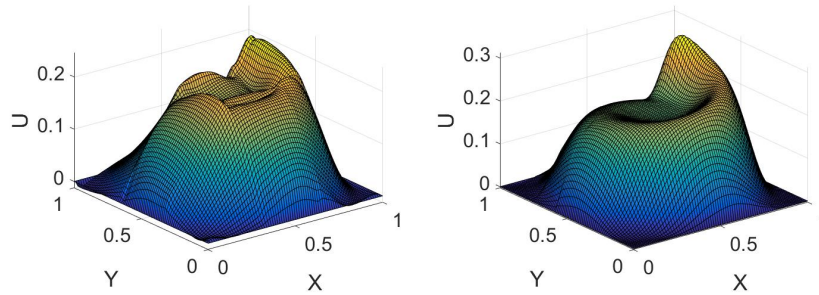
Figure 4.14 – Left, 4 PGDs. Right, 16 PGDs. Both solutions have four modes. Solution of the CRD equation.
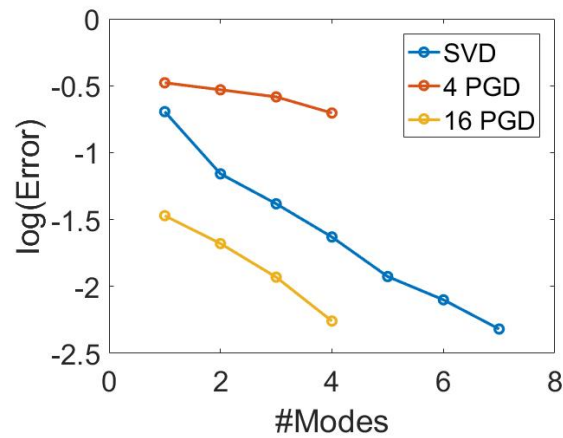

Figure 4.15 – Convergence error for CRD problem in logarithmic scale for different number of modes. Comparison between SVD and two different partitions of the domain.
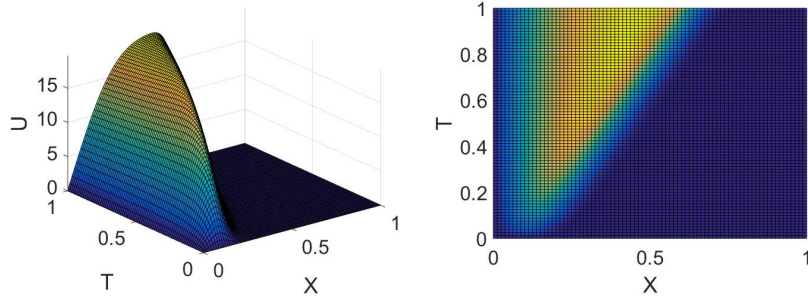
Figure 4.16 – Left, 3D view of $u(x,t)$. Right, planar view of $u(x,t)$ for transient heat transfer equation.

4.1.3.3 Transient Heat Transfer PDE

This very last example is devoted to the analysis of a transient 1D bar heat transfer problem. The weak form associated to the problem is shown in eq. (4.25). Where the diffusive term is set to $\eta = 0.01$.

$$\int_{\Omega} u^* \frac{\partial u}{\partial t} + \eta \frac{\partial u^*}{\partial x} \frac{\partial u^*}{\partial x} \, dxdy = \int_{\Omega} u^* f_3 dxdy \tag{4.25}$$

The domain of study is $\Omega = \Omega_x \times \Omega_t = [0,1] \times [0,1]$. The appropiate set of boundary conditions imposed for the field $u(x,t)$ are $u(x,0) = 0$, $u(0,t) = 0$ and $u(1,t) = 0$. The source term $f_3(x,y)$ is depicted in eq.(4.26) where $\sigma = 0.05$, $v = 0.5$ and $x_0 = 0.1$.

$$f_3(x,y) = \frac{10}{\sigma\sqrt{2\pi}} e^{-\frac{(x-(vy+x_0))^2}{2\sigma^2}} \tag{4.26}$$

Fig. (4.16) depicts the resulting $u(x,y)$ scalar field for the transient heat equation from different perspectives. As it can be seen, source term diffused in the x direction and it is purely convected along the t direction. This kind of PDE creates a boundary layer along the antidiagonal of the space-time domain which is very hard to capture when using standard low rank approximations i.e. POD and classical PGD.

Two different partitions have been tested as shown in fig. (4.17). It is important to highlight that PGDs acting on $x = 0$, $x = 1$ and $t = 0$ are set to zero to satisfy the null Dirichlet boundary conditions in this region of the boundary. However, the PGD acting on the interior nodes of the line $t = 1$ are not set to zero. This fact comes directly from the pure convective behaviour that present the time variable. Indeed, the line $t = 1$ acts like a outflow boundary, thus, no boundary condition should be imposed there to ensure the well-posedness of the problem.

Fig. (4.18) shows the reconstructed solution using 4 modes per local PGD, when the domain has 6 active PGDs (left) and 20 PGDs (right). As it can be seen, both approximations capture the main features of the reference solution. However, the approximation with 6 active PGD presents small oscillations due to the lack of PGD modes to converge to the reference solution.

Fig. (4.19) shows the convergence of the relative error in logarithmic scale for the SVD and the two different partitions proposed for the transient heat equation. It can be stated that both partitions present a faster convergence than the SVD convergence. However, the
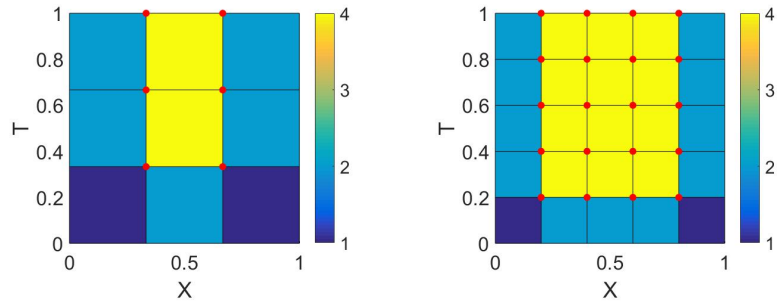
148

Figure 4.17 – Left, domain partioned into 6 PGDs. Right, domain partitioned into 20 PGDs. Red points, centroid of each PGD. Element colour, number of PGDs per element.
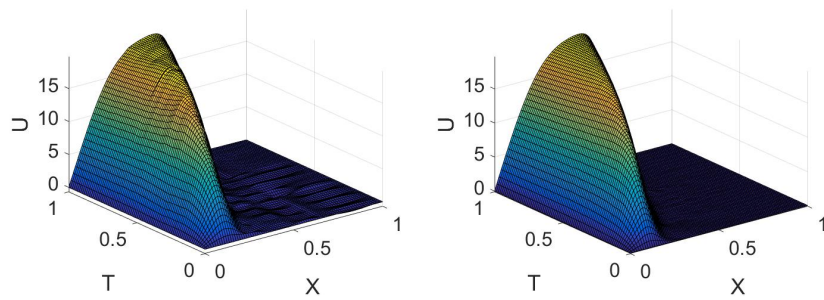

Figure 4.18 – Left, 6 PGDs. Right, 20 PGDs. Both solutions have four modes. Solution of the transient heat equation.
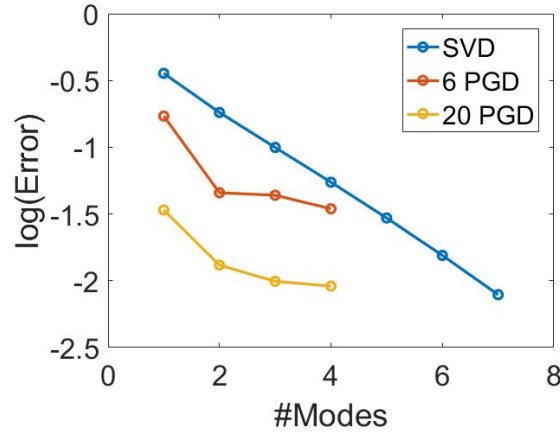
Figure 4.19 – Convergence error for transient heat problem in logarithmic scale for different number of modes. Comparison between SVD and two different partitions of the domain.

convergence slope of the PGD partitions tend to be flatter than the SVD one in the last part of the convergence plot.

4.2 A Multi Scale Proper Generalized Decomposition based on the Partition of Unity

Solutions of partial differential equations could exhibit a multiscale behaviour. Standard discretization techniques are constraint to mesh up to the finest scale in order to predict accurately the response of the system. The multi-PGD methodology presented in section 4.1 is extended to deal with multiscale problems. Indeed, the methodology is based on two main pillars: a generalization of the idea of time domain partition, together with the overlapped multi-PGD framework. By doing that, the proposed methodology naturally inherits continuity between macro subdomains and clear extrapolation to higher dimensional spaces.

4.2.1 Methodology

This section explains the main characteristics of the multi-scale PGD based on the partition of unity. The formulation is introduced for a one-dimensional variable first. Then, the formulation is extended to 2D problems.

Let us assume that a given function, $u(x)$, is the solution of a given partial differential equation. In terms of standard approximation basis, such as finite elements, it could be expressed as

$$u(x) = \sum_{i=1}^{N} N_i(x)u_i, \qquad (4.27)$$

where N stands for the number of degrees of freedom used in the approximation of $u(x)$ or, in other words, the number of nodes in the mesh. Needless to say, $N_i(x)$ represent the

standard finite element shape functions and u_i the nodal value of the sought function, if the employed approximation is interpolant.

However, if the solution presents a multi-scale approach, the mesh has to capture the details of the solution at the finest scale, thus deriving into a prohibitive simulation cost ($N \gg 1$). A possible way to circumvent this issue, within the PU paradigm, is to introduce a dependent variable that captures the solution details associated to the finest scale,

$$u(x) = \sum_{i=1}^{N} N_i(x) u_i \sum_{j=1}^{J} G_j(\tau(x - x_i)) g_j, \qquad (4.28)$$

where x_i is the centroid of the shape function $N_i(x)$, $\tau(x - x_i)$ is a dependent variable which presents an offset based on x_i, $G_j(\tau)$ is the j-th micro-scale shape function and g_j its associated micro-scale degree of freedom. Indeed, the micro scale effects occurring in the compact support of the macro shape function $N_i(x)$ are going to be mapped into a micro scale parent space, τ.

It is worth noting that in Eq. (4.28) macro degrees of freedom (u_i) and micro degrees of freedom (defining the local micro enrichment, g_j) are both unknown, in opposition to usual applications of PU where the enrichment functions are known and given a priori.

Fig. 4.20 shows the shape functions associated to both the macro (top) and the micro scale (bottom). Notice how a two-scale approach presents two meshes related to micro and macro scales, respectively. Indeed, each macro partition of the domain contributes to the parent micro scale and the parent micro space affects equally the macro partition.
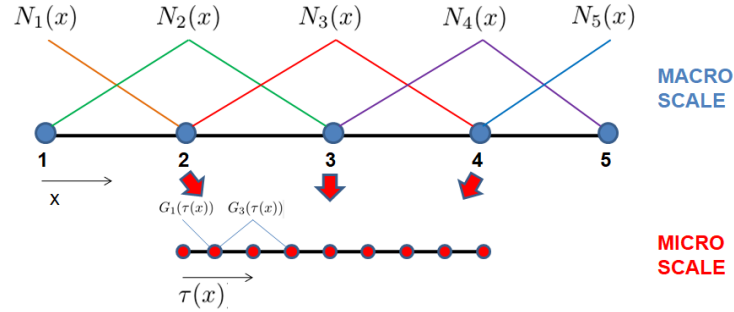
Figure 4.20 – Shape functions of a multi-scale approach. Top, macro shape functions. Bottom, micro shape functions.

Moreover, it is worth mentioning the possibility of rewriting our solution function as $u(x, \tau)$, making it suitable for a PGD-like algorithm[Ryckelynck et al. 2006, ?] but in this case we proceed with a separation of variables as shown in Fig. (4.21). Therefore, the initial one dimensional problem is transformed into a two dimensional problem. It is important to highlight that the macro mesh does not have to capture the solution of the finest scale, since it is handled by the micro scale mesh.

Thus, following a standard PGD rationale[?, ?], the solution is sought in a Greedy manner through of a finite sum of M modal enrichments,

$$u(x, \tau) = \sum_{m=1}^{M} \sum_{i=1}^{N} N_i(x) u_i^m \sum_{j=1}^{J} G_j(\tau(x - x_i)) g_j^m = \sum_{m=1}^{M} \sum_{i=1}^{N} N_i(x) u_i^m \mathbf{G}^T(\tau(x - x_i)) \mathbf{g}^m,$$

$$(4.29)$$
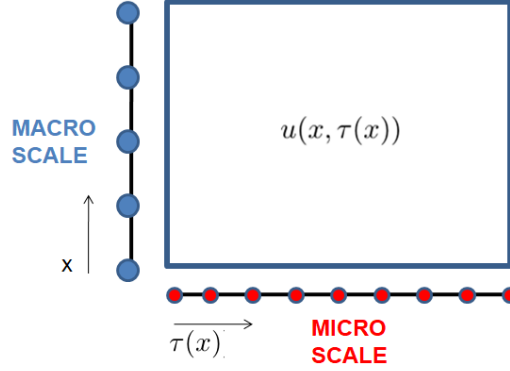
Figure 4.21 – A separation of dependent variables, $u(x, \tau(x))$.

where the super-index m indicates the m-th PGD mode.

The discretization of a variational formulation by means of FEM, for instance, also needs an approximating space for the admissible variation of the field. In this particular case, when computing the M-th mode, we used a standard Galerkin projection

$$w^*(x, \tau) = \sum_{i=1}^{N} N_i(x) u_i^* \mathbf{G}^T(\tau(x - x_i)) \mathbf{g}^M + \sum_{i=1}^{N} N_i(x) u_i^M \mathbf{G}^T(\tau(x - x_i)) \mathbf{g}^*. \qquad (4.30)$$

Nevertheless, SUPG or GLS stabilization can be also implemented in this formulation when dealing with highly convective (parabolic, in general) problems, see also [**?**].

Even if any linearization technique is possible, an alternated direction scheme is chosen in this work to solve the non-linear system associated to the separation of variables. Hence, when solving micro-scale modes, macro-scale admissible variations are set to zero (i.e., $u_i^* = 0$) and, conversely, when solving macro-scale modes, $\mathbf{g}^* = \mathbf{0}$.

Special attention needs to be paid when dealing with the derivatives of the approximation space, viz.

$$\frac{Du(x, \tau)}{Dx} = \frac{\partial u(x, \tau)}{\partial x} + \frac{\partial u(x, \tau)}{\partial \tau} \frac{\partial \tau}{\partial x} = \sum_{m=1}^{M} \sum_{i=1}^{N} \left(\frac{\partial N_i(x)}{\partial x} u_i^m \mathbf{G}^T(\tau) \mathbf{g}^m + N_i(x) u_i^m \frac{\partial \mathbf{G}^T(\tau)}{\partial \tau} \frac{\partial \tau}{\partial x} \mathbf{g}^m \right)$$
$$(4.31)$$

shows the application of the chain rule in order to account for the dependance of $\tau(x - x_i)$. For the sake of simplicity, the dependance of τ with respect to $(x - x_i)$ coordinate will be omitted if there is no risk of confusion.

The same rationale can be easily applied to higher dimensional spaces. Imagine that a 2D case (i.e., $\mathbf{x} = (x, y)$) is approximated by means of a 2D macro mesh and the details of the micro scale are given by one-dimensional modes acting along each spatial direction. Hence, the approximation space will read

$$u(\mathbf{x}, \tau_x, \tau_y) = \sum_{m=1}^{M} \sum_{i=1}^{N} N_i(\mathbf{x}) u_i^m \mathbf{G}^T(\tau_x) \mathbf{g}^m \mathbf{H}^T(\tau_y) \mathbf{h}^m, \qquad (4.32)$$

where $\mathbf{g}^m$ and $\mathbf{h}^m$ are the degrees of freedom of the m-th PGD mode acting on x and y subscales, respectively. Indeed, u_i adopts the same role than in the 1D case, since it will be responsible of weighting the subgrid behavior throughout the domain.
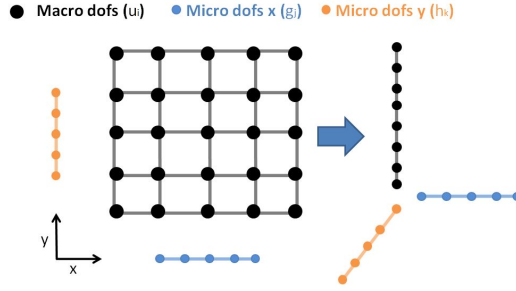
Figure 4.22 – A separation of dependent variables in a 2D space, $u(\mathbf{x}, \tau_x, \tau_y)$.

The separated representation by means of partition of unity-based PGD in higher dimensions is depicted in Fig. 4.22. As it can be noticed, an initial 2D problem is transformed into a 3D problem, where one dimension takes into account the macro variations of the solution, whereas the other two dimensions take into account the subgrid scales taking place along x and y directions, respectively. It is very important to notice that subgrid scales are going to propagate throughout the entire domain, being the macro mesh the responsible of controlling the propagation of the subgrid behavior.

It is also important to highlight that the proposed approach reduces the amount of degrees of freedom required to handle subgrid scales in comparison with standard multigrid methods[?]. Imagine that a 2D domain is partitioned using a coarse regular mesh composed of $D_x \times D_y$ linear elements. Let us assume that each element is refined even more to handle subgrid scales using a mesh of $d_x \times d_y$ linear elements. The typical size of the entire system will be $D_x \times D_y \times d_x^m \times d_y^m$. Nevertheless, the proposed methodology is able to handle subgrid scales at a computer cost proportional to $M((D_x \times D_y) + d_x + d_y)$. Indeed, if the number of modes required to represent the solution does not grow too much (i.e., $M = \mathcal{O}(10)$), computing subgrid scales becomes affordable.

4.2.2 Numerical examples

In this section, several numerical examples based on the multi-scale PGD formulation are tested. The first part shows the convergence of the method for three different one-dimensional cases. The second part shows several examples of the proposed methodology for different two-dimensional cases.

4.2.2.1 One-dimensional numerical examples

In this first section we analyze three different toy problems in one dimension. These include approximation, diffusion and convection problems.

Multiscale approximation of a given function   This first example concerns the approximation of a function that shows multiscale features. Since we deal with an approximation case, there is no governing partial differential equation in this problem. In other words, a given function $f(x)$ is to be approximated by means of a multi-scale approximation $u(x, \tau)$. The weighted residual form associated to this problem reads

$$\int_\Omega w^*(x, \tau) u(x, \tau) dx = \int_\Omega w^*(x, \tau) f(x) dx \quad \forall\ x \in \Omega = [0, 1]. \tag{4.33}$$

In this case the function to approximate is defined as

$$f(x) = \frac{x\cos(8\pi x)}{8\pi}. \qquad (4.34)$$

Fig. 4.23 shows the reconstructed solution (red line) versus the reference solution (blue line) when the macro domain is partitioned using 5 degrees of freedom (black dots), i.e., $N = 5$. The micro domain is partitioned using 80 linear finite elements, i.e., $J = 81$. As it can be noticed, the macro partition coincides exactly with the period of the signal, which is precisely the reason why only 1 mode ($M = 1$) is required to accurately represent the solution. Indeed, the micro scale mode is giving us the cosine signal whereas the macro scale is taking care of the linear growth.
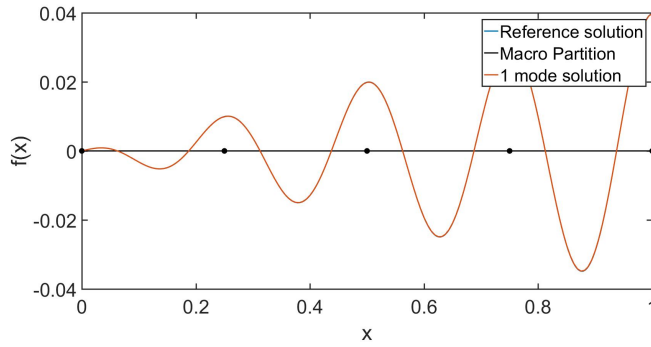

Figure 4.23 – Approximation problem. The reconstructed solution with one single mode is represented by the red line. The reference solution is the blue line. In this case, the macro-domain partition employed 5 degrees of freedom, represented as black dots.

However, the exact period of our signal is not known a priori in most cases. Therefore, convergence of the method when the macro partition does not coincide with the signal period is checked as well. Fig. 4.24 shows the reconstructed solution (red line) against the reference one (blue line) when the macro domain partition does not coincide with the signal period. The top figure, involves a reconstructed solution with one mode, whereas the bottom one involves five modes. It is important to notice that the one-mode solution captures the main trend of the signal. However, there are some regions where the signal is not properly captured. Indeed, the solution involving five modes reproduces the reference signal very accurately.

Fig. 4.25 shows the relative error in logarithmic scale between the reconstructed solution and the reference solution for different number of PGD modes. As it can be noticed, the multiscale PGD converges monotonically.

Diffusion Case The second test problem is a pure diffusion equation. The weak form associated to this problem is

$$\int_\Omega \frac{Dw^*(x,\tau)}{Dx} \frac{Du(x,\tau)}{Dx} dx = \int_\Omega w^*(x,\tau(x))f(x)dx, \quad \forall \ x \in \Omega = [0,10]. \qquad (4.35)$$

The chain rule is required, as in Eq. (4.31), due to the dependence of τ with respect to x. The source term and the two boundary conditions ensuring that the problem is well posed
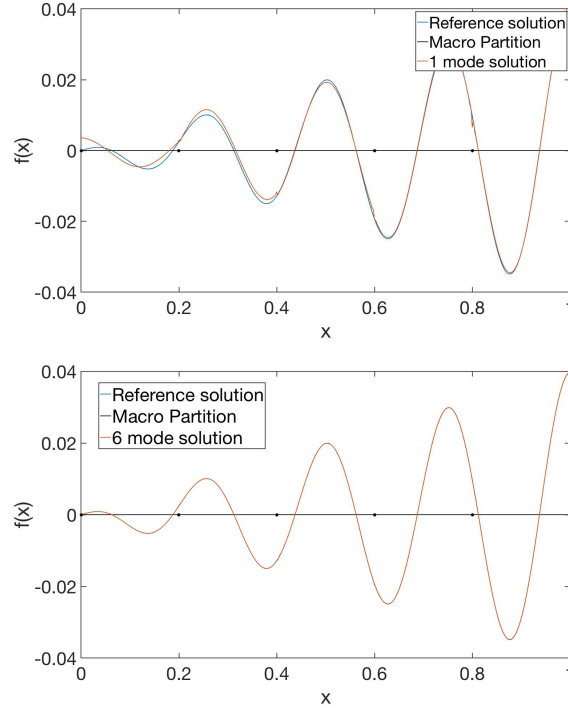
Figure 4.24 – Approximation problem. Top: solution with one single mode. Bottom: 6 modes. The reconstructed solution appears as a red line. The reference solution is represented by a blue line. Macro-domain partition with 6 degrees of freedom, appearing as black dots.

are

$$f(x) = \cos(2\pi x) + \frac{x}{100}, \tag{4.36}$$

$$u(0) = 0, \tag{4.37}$$

$$u(10) = 0. \tag{4.38}$$

Under these conditions, the analytical solution of the problem reads

$$u(x) = \frac{\cos(2\pi x) - 1}{4\pi^2} + \frac{x}{6} - \frac{x^3}{600}. \tag{4.39}$$

Fig. 4.26 compares the reconstructed solution using the first PGD mode against the reference solution. The macro domain has been partitioned using 8 degrees of freedom. As it can be noticed, the first mode already captures the macro behavior, however, extra modes are required to identify the oscillatory behavior of the micro scale. Indeed, the reconstructed solution with 3 modes already captures the oscillations of the micro scale.

Fig. 4.27 shows the convergence the reconstructed solution with respect to the number of PGD modes. Again, results show good convergence properties towards the reference, exact solution.

Convection Case   The third example is a convection problem, whose weak form reads

$$\int_\Omega w^*(x, \tau(x)) \frac{Du(x, \tau(x))}{Dx} dx = \int_\Omega w^*(x, \tau(x)) f(x) dx \quad \forall\ x \in \Omega = [0, 10]. \tag{4.40}$$

155

Figure 4.25 – Relative error of the reconstructed solution with respect to the reference solution as a function of the number of PGD modes for the approximation problem.

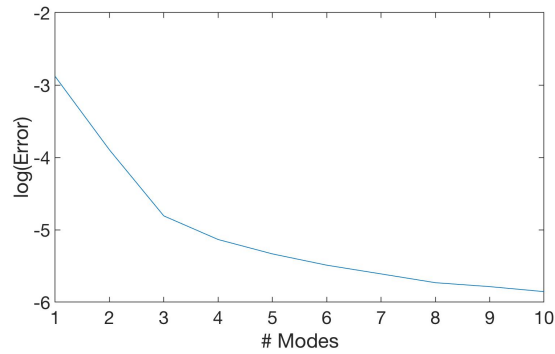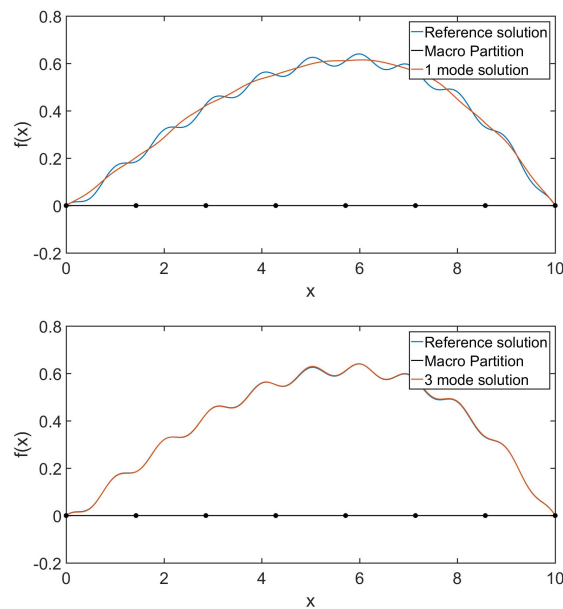

Figure 4.26 – Diffusion problem. Top, 1 mode. Botoom, 3 modes. Reconstructed solution, red line. Reference solution, blue line. Macro domain partition with 8 degrees of freedom, black dots.
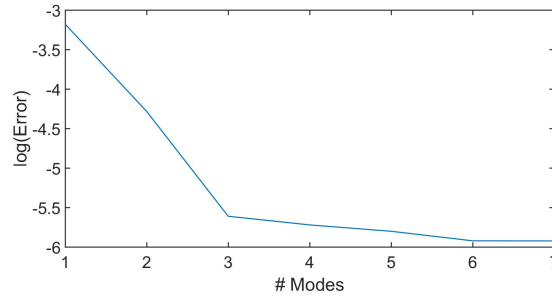
Figure 4.27 – Relative error of the reconstructed solution with respect to the reference solution as a number of PGD modes for the diffusion case.

It is well-known that special attention has to be paid when dealing with convection-dominated equations to ensure convergence. A standard streamline-upwind Petrov-Galerkin (SUPG) is implemented in this particular case[S. Fernandez 2004]. Therefore, the test function is defined as

$$w^*(x, \tau(x)) = u^*(x, \tau(x)) + \beta \frac{Du^*(x, \tau(x))}{Dx}, \tag{4.41}$$

where β is a numerical coefficient controlling the stabilization of the numerical scheme. It is important to notice that the numerical scheme remains consistent since the test function affects both sides of Eq. (4.40).

The source term of the problem is

$$f(x) = x \sin(12\pi x) + \frac{x}{30}, \tag{4.42}$$

while the initial condition 8left boundary) is taken as

$$u(0) = 0. \tag{4.43}$$

In that case, the analytical solution associated to this initial value problem is

$$u(x) = \frac{\sin(12\pi x)}{144\pi^2} - \frac{x \cos(12\pi x)}{12\pi} + \frac{x^2}{60}. \tag{4.44}$$

Fig. 4.28 shows the reconstructed solution involving one (top) and seven (bottom) modes versus the reference one. The macro domain has been partitioned using 9 degrees of freedom. As it can be noticed, the reconstructed solution is far from the reference one, when only one mode is involved. However, the PGD algorithm converges towards the reference solution when more modes are added into the approximation.

In turn, Fig. 4.29 shows the convergence of the multi-scale PGD algorithm for the 1D convection case. The method converges well, reaching a relative error of 10^{-4} when using 7 modes. It is important to notice that the error seems to stagnate after 5 PGD modes. This stagnation is linked with the discretization of the macro-micro domains. Indeed, an approximation is already done when macro-micro domains are discretized, being a possible source of difference between reference and reconstructed solutions.

4.2.2.2 Two dimensional numerical examples

This section extends the analysis of the developed methodology to three different two-dimensional cases: approximation, diffusion and convection-reaction-diffusion problems.

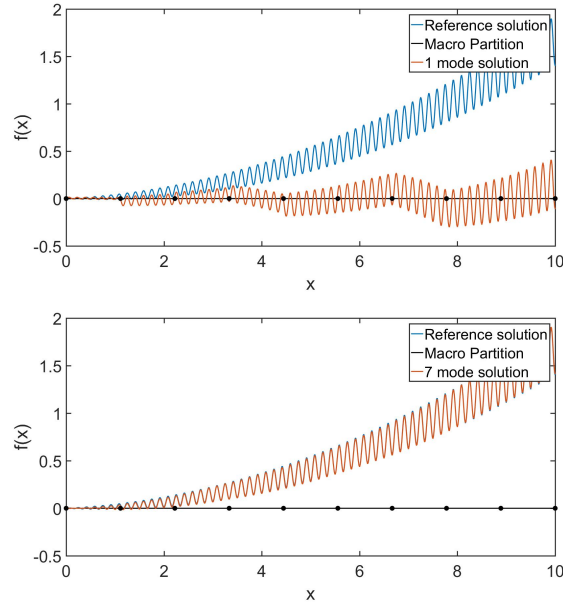Figure 4.28 – Convection case. Top, 1 mode. Botoom, 7 modes. The reconstructed solution is represented as a red line, while the reference solution appears in blue line. Macro domain partition with 10 degrees of freedom, represented as black dots.

Figure 4.29 – Relative error of the reconstructed solution with respect to the reference solution as a number of PGD modes for the convection case.
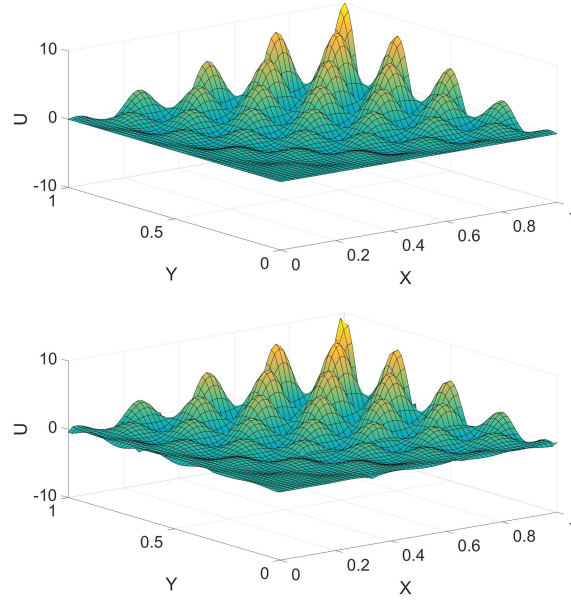
Figure 4.30 – 2D Approximation case. Top: reference solution. Bottom: reconstructed solution. Macro domain partitioned with $5 \times 5 = 25$ degrees of freedom.

2D approximation problem Again, we consider the multiscale approximation of a given function $f(\mathbf{x})$ by a sought function $u(\mathbf{x})$. The weighted residual (Bubnov-Galerkin) form of this problem is

$$\int_\Omega w^*(\mathbf{x}, \tau_x, \tau_y) u(\mathbf{x}, \tau_x, \tau_y) d\mathbf{x} = \int_\Omega w^*(\mathbf{x}, \tau_x, \tau_y) f(\mathbf{x}) d\mathbf{x} \quad \forall \, \mathbf{x} \in \Omega = [0,1]^2 \qquad (4.45)$$

In this case we consider a function

$$f(\mathbf{x}) = x \cos(8\pi x) y \cos(8\pi y). \qquad (4.46)$$

Note that no boundary conditions need to be imposed for this problem.

Fig. 4.30 shows the analytical solution (right) versus the reconstructed solution using the multi-scale PGD algorithm with 9 modes. The macro domain has been partitioned using a mesh of 25 degrees of freedom, since each spatial direction has been partitioned using 5 nodes. As it can be clearly seen, the reconstructed solution captures the main features of the reference solution. The small differences are due to the fact that the partition of the macro domain does not coincide with the period of the signal. Therefore, extra modes will be required to alleviate these small oscillations.

Fig. 4.31 shows the convergence plot associated to the 2D approximation case as a function of the modal enrichments of the solution. Note the monotone convergence of the solution.

2D diffusion case The weak form associated to this diffusion problem is

$$\int_\Omega \frac{Dw^*(\mathbf{x}, \tau_x, \tau_y)}{D\mathbf{x}} \cdot \frac{Du(\mathbf{x}, \tau_x, \tau_y)}{D\mathbf{x}} d\mathbf{x} = \int_\Omega w^*(\mathbf{x}, \tau_x, \tau_y) f(\mathbf{x}) d\mathbf{x} \quad \forall \, \mathbf{x} \in \Omega = [0,1]^2. \quad (4.47)$$
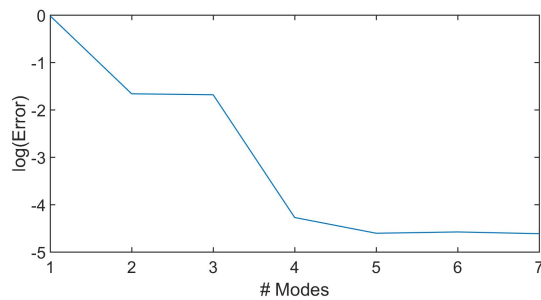
159

Figure 4.31 – Relative error of the reconstructed solution with respect to the reference solution as a number of PGD modes for the 2D approximation case.
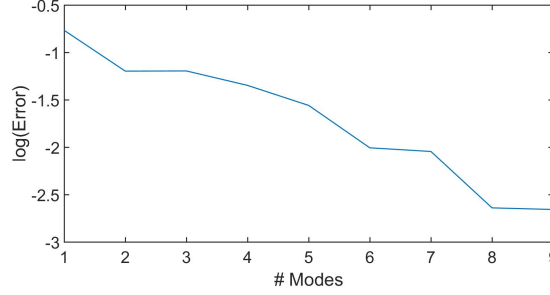
with boundary conditions

$$u(0, y) = u(1, y) = u(x, 0) = u(x, 1) = 0. \tag{4.48}$$

The chain rule is required, as shown in Eq. (4.31), due to the dependence of τ_x and τ_y with respect to x and y, respectively. The source term f considered in this case was

$$f(\mathbf{x}) = 2w^2 xy \sin(wx)\sin(wy) - 2wx\sin(wx)\cos(wy) - 2wy\cos(wx)\sin(wy). \tag{4.49}$$

In this particular case, the angular velocity is set to $w = 20\pi$. In that situation, the analytical solution is given by

$$u(x, y) = x\sin(wx)y\sin(wy). \tag{4.50}$$

It is important to highlight the treatment of the boundary conditions in order to solve the diffusion problem. Vanishing Dirichlet boundary conditions need to be imposed on the boundary, so that all macro degrees of freedom placed at the boundary are fixed to zero. To enforce non-vanishing Dirichlet boundary conditions, the first modes will satisfy the Dirichlet boundary conditions whereas the subsequent modes will be computed with vanishing boundary conditions, following the standard rationale of the PGD[?].

Fig. 4.32 shows the reference solution for the diffusion case (top) versus the reconstructed solution using 5 modes of the PGD. The macro domain has been partitioned using a mesh of 4 nodes per direction but since the nodes at the boundary are set to zero, the final degrees of freedom are $2 \times 2 = 4$. Note that there is no perceivable difference between the reference solution and the reconstructed one.

2D convection-diffusion problem   The weak form associated to this diffusion problem is

$$\int_\Omega w^*(\mathbf{x}, \tau_x, \tau_y)\frac{Du(\mathbf{x}, \tau_x, \tau_y)}{Dy}d\mathbf{x} + \int_\mathbf{x} \frac{Dw^*(\mathbf{x}, \tau_x, \tau_y)}{Dx}\frac{Du(\mathbf{x}, \tau_x, \tau_y)}{Dx}d\mathbf{x} = \int_\Omega w^*(\mathbf{x}, \tau_x, \tau_y)f(\mathbf{x})d\mathbf{x}, \quad \forall\, \mathbf{x} \in \Omega = [0, 1]^2 \tag{4.51}$$

Note that along the y direction there is a pure convection phenomenon, whereas the x direction presents pure diffusion. Hence, the nature of the PDE forces to impose boundary conditions at both ends of the x interval, whereas only initial boundary conditions must be imposed in the y direction. The source term reads in this case

$$f(\mathbf{x}) = (16w^2 y^2 + 2y)\sin(4wx) + (64w^2 - w)\sin(8wx)\sin(wy) \tag{4.52}$$
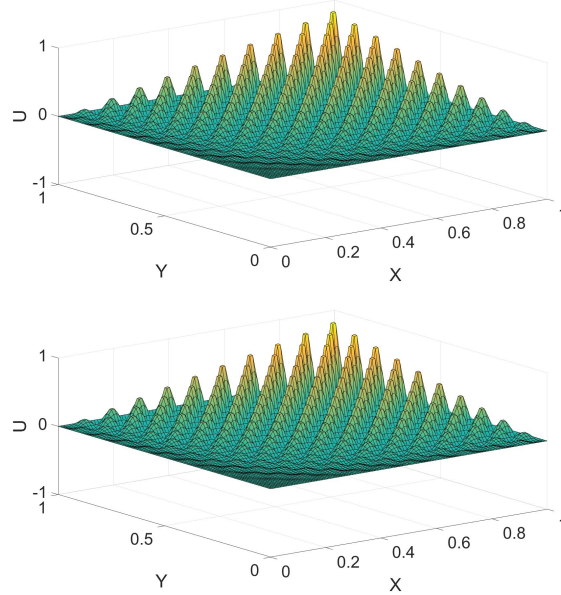
Figure 4.32 – 2D Diffusion case. Top: reference solution. Bottom: reconstructed solution with 5 modes. Macro domain partitioned with $2 \times 2 = 4$ degrees of freedom.
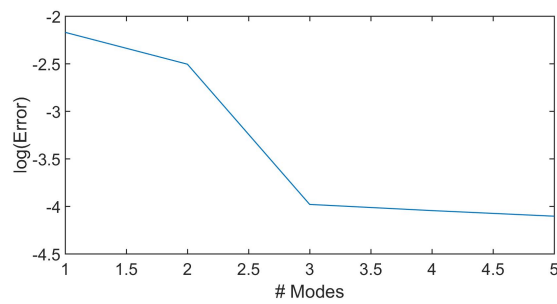
Figure 4.33 – Relative error of the reconstructed solution with respect to the reference solution as a number of PGD modes for the 2D diffusion case.
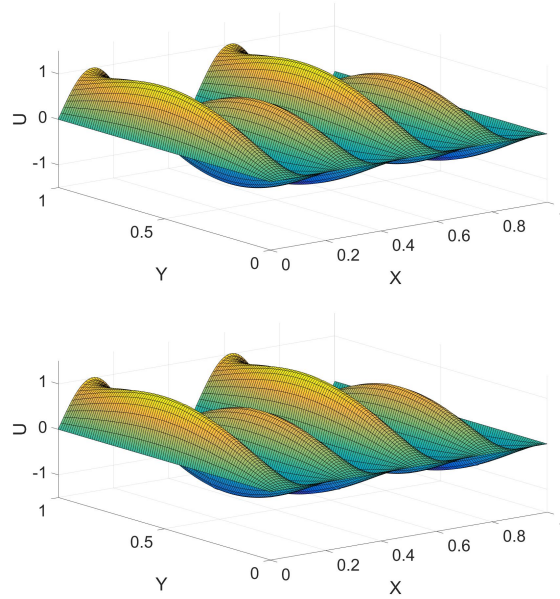
Figure 4.34 – 2D Convection-Diffusion case. Top: reference solution. Bottom: reconstructed solution with 5 modes. Macro domain partitioned with $3 \times 4 = 12$ degrees of freedom.

while the set of boundary conditions ensuring the well-posedness of the problem reads

$$u(0,y) = u(1,y) = u(x,0) = 0. \tag{4.53}$$

In this case the angular velocity is set to $w = \pi$.

Under this set of conditions, the partial differential equation (4.51) admits a unique solution given by

$$u(\mathbf{x}) = \sin(4wx)y^2 + \sin(8wx)\sin(wy). \tag{4.54}$$

Fig. (4.34) shows the reference solution (top) versus the reconstructed solution involving 5 modes (bottom). The macro domain has been partitioned using $3 \times 4 = 12$ degrees of freedom. As it can be noticed, the reconstructed solution already captures the main features of the reference solution. It is important to notice how the reconstructed solution is able to reproduce the set of homogeneous boundary conditions imposed in the 3 sides of the squared domain.

Fig. 4.35 shows the relative error versus the number of modes involved in the reconstructed solution. As it can be seen, the error decreases with respect to the number of modes, showing good convergence properties even for convection diffusion problems.

Time Multi-Scale applied to a discrete system of equations   The main aim of this example is to show the capabilities of the algorithm to predict the response of a transient system of equations arising from a standard FEM discretization in space. The strong form of the problem reads as shown in eq. (4.55).

$$\frac{\partial u(x,t)}{\partial t} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t) \tag{4.55}$$
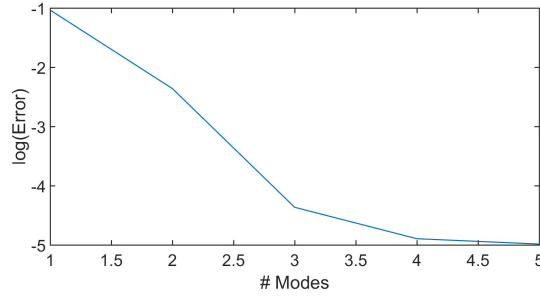
Figure 4.35 – Relative error of the reconstructed solution with respect to the reference solution as a number of PGD modes for the 2D convection diffusion case.

Indeed, the problem analyzed in this section makes reference to a 1D transient thermal problem subjected to the boundary conditions and external source term shown in eq. (4.56).

$$
\begin{aligned}
f(x,t) &= (4\pi^2(sin(t) + sin(50t)) + cos(t) + 50cos(50t))sin(2\pi x) \\
u(x,0) &= 0 \\
u(0,t) &= u(1,t) = 0
\end{aligned}
\tag{4.56}
$$

This partial differential equation under this set of boundary conditions admits an analytical solution given in eq. (4.57). It is clear to see that this solution presents a two scale behaviour in the time domain.

$$
u(x,t) = (sin(t) + sin(50t))sin(2\pi x)
\tag{4.57}
$$

If the spatial 1D bar is discretized using 80 linear elements equally spaced, it gives the discrete system of equations (4.58), where multi-scale is applied only in time.

$$
\mathbf{M}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{f}(t) \in t = [0, T]
\tag{4.58}
$$

Fig. 4.36 shows the reference solution (left) and the reconstructed solution (right) using 1 mode for the discretized transient thermal problem. As it can be noticed, the obtained solution is in good agreement even when involving only one mode. This fact is not surprinsing since the analytical solution is composed only by one separated mode. Indeed, Fig. (4.37) shows that only one mode involves $1e-5$ of relative error with respect to the analytical solution.

4.2.3 Conclusions

A novel algorithm able to solve multi-scale problems in an efficient way is proposed in this work. The algorithm combines a macro partition of the domain which is enriched with unidimensional modes following standard PGD rationale. The convergence of the algorithm is proven to be very effective for 1D and 2D problems, involving algebraic, diffusion and convection phenomena.

163
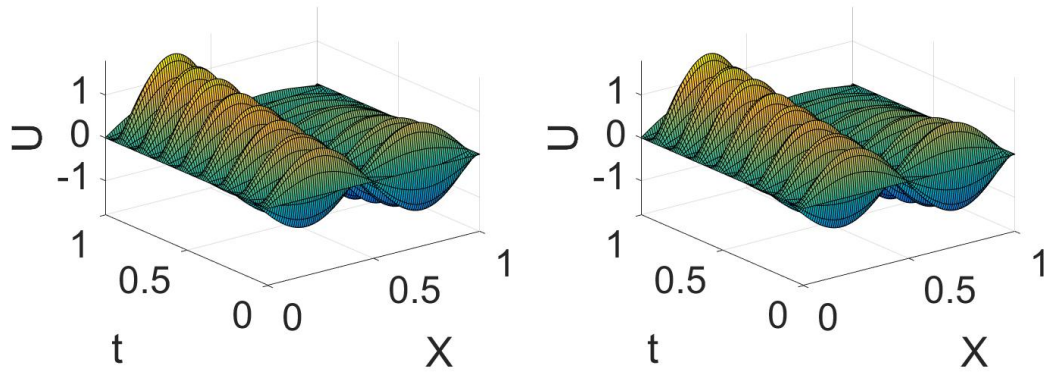
Figure 4.36 – Transient thermal problem. Reference solution, left. Reconstructed solution with 1 mode, right. Temporal domain partitioned with 4 degrees of freedom.
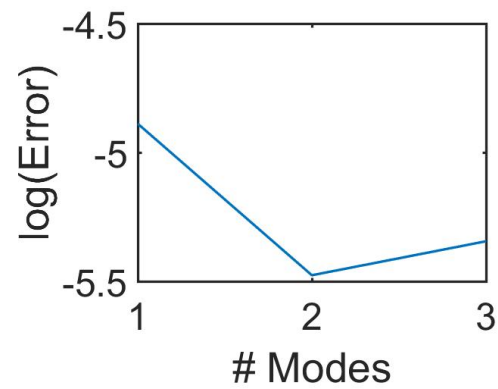
Figure 4.37 – Convergence of the transient thermal problem for different number of modes.

The methodology renders good results even when the macro domain partition does not involve a pure periodical signal in the micro scale. The extension of the algorithm to higher dimensional spaces is our current line of research just like the extension of the methodology to non-rectangular domains.

Applications of PGD In-Plane-Out-Of-Plane Separations

Abstract Many electrical and structural components are constituted of a stacking of multiple thin layers with different electromagnetic, mechanical and thermal properties. The introduction of appropriate kinematic and mechanic hypotheses allow to simplify the general 3D problem into a 2D involving just the in-plane coordinates. This was the route, for instance, employed for deriving beam, plate and shell theories in solid mechanics, that were extended later to many other physics, like flows in narrow gaps or thermal, among many others. However, in many cases, when addressing complex coupled physics the validity of hypotheses able to reduce models from 3D to 2D becomes doubtful and consequently in order to ensure accurate results 3D discretizations seem compulsory. To circumvent the numerical difficulties that such a rich description imply, an in-plane-out-of-plane separated representation with the aim of computing fully 3D solutions as a sequence of 2D problems defined in the plane coordinates and others (1D) in the thickness coordinate was proposed in [Bognet *et al.* 2012, Bognet *et al.* 2014].

Contents

This chapter portraits novel applications of the so-called PGD In-Plane-Out-Of-Plane separated representations. Section 5.1 proposes of an efficient in-plane-out-of-plane separated representation of the double-curl formulation of Maxwell equations able to address thin-layer laminates while ensuring the continuity and discontinuity of the tangential and normal electric field components respectively at the plies interface. For the sake of completeness, the main ingredients behind such a electromagnetic formulation are detailed as well.

Section 5.2 extends the in-plane-out-of-plane separated representations successfully used for addressing fully 3D model solutions defined in plate-like domain, to dynamics. Herein, we introduce a new efficient hybrid explicit in-plane implicit out-of-plane time integration scheme for dynamic problems defined in plate-like domains that allows to compute 3D solutions with the stability constraint exclusively determined by the coarser in-plane discretization. Particularly, the mesh employed for discretizing the out-of-plane dimension (thickness) determines the maximum time-step ensuring stability when using an explicit time integration scheme.

Finally, section 5.3 derives another application for unidirectional composite prepegs. In this particular case, the thin composite ply is modeled by using an anisotropic incompressible viscous flow which becomes inextensible along the fiber direction. An extensive analysis on the numerical stability of the separated representation is performed since the flow is subjected to two kinematic constraints, incompressibility and inextensibility, which may lead to inf-sup (LBB) conditions.

These topics are addressed in three published papers:

- H. Tertrais, R. Ibáñez, A. Barasinski, C. Ghnatios, F. Chinesta, *On the Proper Generalized Decomposition applied to microwave processes involving multilayered components.* Mathematics and Computers in Simulation, **156**, 347-363, 2019.

- G. Quaranta, B. Bognet, R. Ibáñez, A. Tramecon, E. Haug, 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 and Structures, **210**, 135-144, 2018.

- R. Ibáñez, E. Abisset-Chavanne, F. Chinesta, A. Huerta *Simulating squeeze flows in multiaxial laminates: towards fully 3D mixed formulations.* International Journal of Material Forming, **210**, 653-669, 2017.

5.1 On the Proper Generalized Decomposition applied to microwave processes involving multilayered components

The main objective of this section is to provide a numerical algorithm able to address a high fidelity resolution of the Maxwell's equations in its double rotational formulation for thin geometries. Subsection 5.1.1 gives an overview of the electromagnetic formulation. Subsection 5.1.2 shows how the in-plane-out-of-plane separated representation applies to the electromagnetic field together with particular interface conditions which must be satisfied at the interface between plies. Numerical results based on both a 3-ply and 29-ply laminate are shown in subsection 5.1.4.

5.1.1 Electromagnetic formulation

The usual approach when solving a general electromagnetic problem with the finite element method is considering edge elements [Nedelec 1980] with the double-curl formulation of Maxwell equations. The use of edge elements allow efficiently circumventing the main problems of FEM applied to electromagnetic models [Jin 2002], in particular they produce spurious-free solutions and ensure the normal discontinuity and tangential continuity between different media, however some disadvantages have been also pointed out [Mur 1994, Mur 1998], concerning the ill-conditioning of discrete systems when the number of degrees of freedom $- dof -$ increases. Some solutions were proposed for circumventing such issue, as for example the introduction of Lagrange multipliers, with the associate *dof* increase.

Many authors preferred the use of nodal-regularized formulations to avoid spurious solutions [Hazard & Lenoir 996], while proposing ad-hoc solutions for accounting the transfer conditions at the material interfaces, as implemented in the ERMES software [Otin 2013], consisting of duplicating the nodes located at the interfaces, for approximating the discontinuous field while enforcing the jump condition. Moreover, in these formulations a second regularization is required for addressing field singularities.

Because in the case of multilayered laminates the interfaces coincide with constant values of the out-of-plane coordinate (the thickness), implementing a discontinuous approximation within the in-plane-out-of-plane separated representation involved in the PGD seems quite simple. Thus, the modeling framework that we are considering consists of standard approximations (the use of advanced discretizations based on edge elements into the PGD framework constitutes a work in progress) combined with a regularized formulation, with an ad-hoc treatment of interface transfer conditions.

The double-curl formulation is derived from the Maxwell's equations in the frequency space, that in absence of current density in the laminate, reads

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{E} \right) - \omega^2 \epsilon \mathbf{E} = 0, \quad \text{in } \Omega \subset \mathbb{R}^3 \tag{5.1}$$

with the complex permittivity ϵ given by

$$\epsilon = \epsilon_r - i \frac{\sigma}{\omega}, \tag{5.2}$$

and where μ, ϵ_r and σ represent the usual magnetic permeability, the electric permittivity and the conductivity respectively.

The previous equation is complemented with adequate boundary conditions. Without loss of generality we are assuming in what follows Dirichlet boundary conditions in the whole domain boundary $\partial\Omega$

$$\mathbf{n} \times \mathbf{E} = \mathbf{E}_g^t, \quad \text{in } \partial\Omega \tag{5.3}$$

where $\mathbf{n}$ refers to the unit outwards vector defined on the domain boundary. In the previous expressions $\mathbf{E}_g^t$ is the prescribed electric field (assumed known) on the domain boundary, tangent to the boundary as Eq. (5.3) expresses.

The weighted residual weak form is obtained by multiplying (5.1) by the test function $\mathbf{E}^*$ (in fact by its conjugate, $\overline{\mathbf{E}}^*$, to define properly scalar products being the electric field a complex field, i.e. $\mathbf{E} = \mathbf{E}_r + i\mathbf{E}_i$), and then integrating by parts, to obtain

$$\int_\Omega \frac{1}{\mu}(\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{E}}^*) \, d\mathbf{x} - \omega^2 \int_\Omega \epsilon\mathbf{E} \cdot \overline{\mathbf{E}}^* \, d\mathbf{x} -$$

$$\int_{\partial\Omega} \frac{1}{\mu}(\nabla \times \mathbf{E}) \cdot (\mathbf{n} \times \overline{\mathbf{E}}^*) \, d\mathbf{x} \; = \; 0, \tag{5.4}$$

for all test function $\overline{\mathbf{E}}^*$ regular enough.

In the previous expression the boundary integral can be removed if the test function is assumed verifying $\mathbf{n} \times \overline{\mathbf{E}}^* = 0$ on $\partial\Omega$ where Dirichet boundary conditions are enforced, i.e. $\mathbf{n} \times \mathbf{E}$ on $\partial\Omega$. As previously indicated, in what follows for the sake of simplicity we are assuming Dirichlet boundary conditions on the whole domain boundary and then the weak form reduces to

$$\int_\Omega \frac{1}{\mu}(\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{E}}^*) \, d\mathbf{x} - \omega^2 \int_\Omega \epsilon\mathbf{E} \cdot \overline{\mathbf{E}}^* \, d\mathbf{x} = 0 \tag{5.5}$$

However, it is well known that the weak form (5.5) produces spurious solutions because even if Eq. (5.1) ensures the verification of the Gauss equation $\nabla \cdot (\epsilon\mathbf{E}) = 0$, its discrete counterpart after approximating the different fields implied in the weak form (5.5) does not ensure the fulfillment of the Gauss equation. Thus, a regularization is compulsory for avoiding these spurious solutions. In what follows we consider the regularized form [Otin 2013]

$$\nabla \times \left(\frac{1}{\mu}\nabla \times \mathbf{E}\right) - \overline{\epsilon}\nabla \left(\frac{1}{\overline{\epsilon}\epsilon\mu}\nabla \cdot (\epsilon\mathbf{E})\right) - \omega^2\epsilon\mathbf{E} = 0, \tag{5.6}$$

whose (regularized) weak form when Dirichlet boundary conditions apply on the wall domain boundary, reads

$$\int_\Omega \frac{1}{\mu}(\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{E}}^*) \, d\mathbf{x} - \omega^2 \int_\Omega \epsilon\mathbf{E} \cdot \overline{\mathbf{E}}^* \, d\mathbf{x} +$$

$$\int_\Omega \frac{\tau}{\overline{\epsilon}\epsilon\mu} (\nabla \cdot (\epsilon\mathbf{E})) \; (\nabla \cdot (\overline{\epsilon}\overline{\mathbf{E}}^*)) \, d\mathbf{x} \; - \int_{\partial\Omega} \frac{\tau}{\overline{\epsilon}\epsilon\mu}(\nabla \cdot (\epsilon\mathbf{E}) \, (\mathbf{n} \cdot (\overline{\epsilon}\overline{\mathbf{E}}^*))) \, d\mathbf{x} \; = \; 0, \tag{5.7}$$

where τ is the regularization coefficient proposed in [Otin 2010] and that according to that reference is taken with a unit value everywhere except at the interfaces where it vanishes.

5.1.2   In-plane-out-of-plane separated representation

To ensure a high enough resolution of the electric field along the component thickness to represent the multilayered structure, we consider an in-plane-out-of-plane separated representation in $\Omega = \Omega_p \times \Omega_t$, with $\Omega_p \subset \mathbb{R}^2$ and $\Omega_t \in \mathbb{R}$. Each point $\mathbf{x} \in \Omega$ is decomposed in its plane component $(x, y) \in \Omega_p$ and its out-of-plane $z \in \Omega_t$.

5.1.2.1 Separated representation

The electric field is expressed as

$$\mathbf{E}(x,y,z) \approx \sum_{i=1}^{N} \mathbf{P}_i(x,y) \circ \mathbf{T}_i(z) = \left(\begin{array}{c} \sum_{i=1}^{N} P_i^x(x,y) \cdot T_i^x(z) \\ \sum_{i=1}^{N} P_i^y(x,y) \cdot T_i^y(z) \\ \sum_{i=1}^{N} P_i^z(x,y) \cdot T_i^z(z) \end{array} \right), \tag{5.8}$$

where "$\circ$" refers to the Hadamard product. The different in-plane and out-of-plane functions, P_i and T_i respectively, are complex and then they involve a real and imaginary parts.

The previous separated representation leads to a separated representation of its derivatives according to

$$\left(\begin{array}{ccc} \frac{\partial E_x}{\partial x} & \frac{\partial E_x}{\partial y} & \frac{\partial E_x}{\partial z} \\ \frac{\partial E_y}{\partial x} & \frac{\partial E_y}{\partial y} & \frac{\partial E_y}{\partial z} \\ \frac{\partial E_z}{\partial x} & \frac{\partial E_z}{\partial y} & \frac{\partial E_e}{\partial z} \end{array} \right) \approx \sum_{i=1}^{N} \left(\begin{array}{ccc} \frac{\partial P_i^x}{\partial x} & \frac{\partial P_i^x}{\partial y} & P_i^x \\ \frac{\partial P_i^y}{\partial x} & \frac{\partial P_i^y}{\partial y} & P_i^y \\ \frac{\partial P_i^z}{\partial x} & \frac{\partial P_i^z}{\partial y} & P_i^z \end{array} \right) \circ \left(\begin{array}{ccc} T_i^x & T_i^x & \frac{\partial T_i^x}{\partial z} \\ T_i^y & T_i^y & \frac{\partial T_i^y}{\partial z} \\ T_i^z & T_i^z & \frac{\partial T_i^z}{\partial z} \end{array} \right) =$$

$$\sum_{i=1}^{N} \mathbb{P}_i(x,y) \circ \mathbb{T}_i(z), \tag{5.9}$$

allowing the separated representation of all the differential operators appearing within the regularized weak form (5.7).

However, the separated representation just proposed requires that all the model parameters accept a similar separated representation.

Consider the laminate composed of $\mathcal{P}$ layers, each one having uniform properties inside. If H is the total laminate thickness, and assuming for the sake of simplicity and without loss of generality that all the plies have the same thickness h, it results $h = \frac{H}{\mathcal{P}}$. Now, we introduce the characteristic function of each ply $\chi_i(z)$, $i = 1, \cdots, \mathcal{P}$:

$$\chi_i(z) = \left\{ \begin{array}{ll} 1 & \text{if } (i-1)h \leq z < ih \\ 0 & \text{elsewehere} \end{array} \right. , \tag{5.10}$$

that allows expressing the different properties in the separated form

$$\left\{ \begin{array}{l} \mu(x,y,z) = \sum_{i=1}^{\mathcal{P}} \mu_i \cdot \chi_i(z) \\ \epsilon(x,y,z) = \sum_{i=1}^{\mathcal{P}} \epsilon_i \cdot \chi_i(z) \end{array} \right. , \tag{5.11}$$

being μ_i and ϵ_i the permeability and complex permittivity (the one that involves the permittivity and conductivity) in the i-layer.

5.1.2.2 Functional approximation

Now, before calculating the different functions involved in the approximation of the electric field, we must approximate them. Because it is assumed that the only heterogeneity applies along the thickness direction when moving from one ply to its neighbor, we can assume a standard continuous nodal approximation of fields E_x and E_y that moreover are continuous

at the ply interfaces. For E_z the situation is a bit different, because it is continuous in the plane but discontinuous across the ply interfaces, where the field jump reads

$$\epsilon_j E_z(z_j^-) = \epsilon_{j+1} E_z(z_j^+), \tag{5.12}$$

where $j \geq 1$ denotes the common interface between plies j and $j+1$, z_j its out-of-plane coordinate, $z_j = jh$, and $E_z(z_j^-)$ and $E_z(z_j^+)$ denote the electric field at both sides of the interface, i.e. $z_j^- = z_j - \nu$ and $z_j^+ = z_j + \nu$, with ν a small enough coefficient ensuring $\nu \ll h$. Discontinuities within the PGD framework were addressed in [Giner et $al.$ 2013, Bur et $al.$ 2016] in mechanical and thermal problems.

Here we consider continuous bilinear quadrilaterals, Q1 for approximating functions depending on the in-plane coordinates, i.e. $P_i^x(x,y)$, $P_i^y(x,y)$ and $P_i^z(x,y)$. In what respect the functions depending on the out-of-plane coordinates we consider linear continuous 1D finite elements for approximating functions $T_i^x(z)$ and $T_i^y(z)$, ensuring the continuity of $E_x(x,y,z)$ and $E_y(x,y,z)$, however in order to enforce the discontinuity of $E_z(x,y,z)$ across the ply interfaces, the nodes of the one-dimensional mesh attached to Ω_z located at the ply interfaces and used for approximating T_i^z, are duplicated as proposed in [Paulsen et $al.$ 1987, Paulsen et $al.$ 1988, Boyse et $al.$ 1992, Otin 2013]. This simple choice ensures the continuity of E_z in the plane and its discontinuity across the ply interfaces.

5.1.3 PGD-based discretization

The separated representation construction proceeds by computing a term of the sum at each iteration. Assuming that the first $n-1$ modes (terms of the finite sum) of the solution were already computed, $\mathbf{E}^{n-1}(x,y,z)$ with $n \geq 1$, the solution enrichment reads:

$$\mathbf{E}^n(x,y,z) = \mathbf{E}^{n-1}(x,y,z) + \mathbf{P}^n(x,y) \circ \mathbf{T}^n(z), \tag{5.13}$$

where both vectors $\mathbf{P}^n$ and $\mathbf{T}^n$ containing functions P_i^n and T_i^n ($i = 1,2,3$) depending on (x,y) and z respectively, are unknown at the present iteration. The test function $\overline{\mathbf{E}}^*$ reads $\overline{\mathbf{E}}^* = \mathbf{P}^* \circ \overline{\mathbf{T}}^n + \overline{\mathbf{P}}^n \circ \mathbf{T}^*$.

The introduction of Eq. (5.13) into (5.7) results in a non-linear problem. We proceed by considering the simplest linearization strategy, an alternated directions fixed point algorithm, that proceeds by calculating $\mathbf{P}^{n,k}$ from $\mathbf{T}^{n,k-1}$ and then by updating $\mathbf{T}^{n,k}$ from the just calculated $\mathbf{P}^{n,k}$ where k refers to the step of the non-linear solver. The iteration procedure continues until convergence, that is, until reaching the fixed point $\|\mathbf{P}^{n,k} \circ \mathbf{T}^{n,k} - \mathbf{P}^{n,k-1} \circ \mathbf{T}^{n,k-1}\| < \nu$ (ν being a small enough coefficient), that results in the searched functions $\mathbf{P}^{n,k} \rightarrow \mathbf{P}^n$ and $\mathbf{T}^{n,k} \rightarrow \mathbf{T}^n$. Then, the enrichment step continues by looking for the next mode $\mathbf{P}^{n+1} \circ \mathbf{T}^{n+1}$. The enrichment stops when the model residual becomes small enough.

When $\mathbf{T}^n$ is assumed known, we consider the test function $\overline{\mathbf{E}}^\star$ given by $\mathbf{P}^\star \circ \overline{\mathbf{T}}^n$. By introducing the trial and test functions into the weak form and then integrating in Ω_t because all the functions depending on the thickness coordinate are known, we obtain a 2D weak formulation defined in Ω_p whose discretization (by using a standard discretization strategy, e.g. finite elements) allows computing $\mathbf{P}^n$.

Analogously, when $\mathbf{P}^n$ is assumed known, the test function $\overline{\mathbf{E}}^\star$ is given by $\overline{\mathbf{P}}^n \circ \mathbf{T}^\star$. By introducing the trial and test functions into the weak form and then integrating in Ω_p because all the functions depending on the in-plane coordinates (x,y) are at present known, we obtain a 1D weak formulation defined in Ω_t whose discretization (using any technique for solving standard ODE equations) allows computing $\mathbf{T}^n$.

As discussed in [Bognet *et al.* 2012] this separated representation allows computing 3D solutions while keeping a computational complexity characteristic of 2D solution procedures. If we consider a hexahedral domain discretized using a regular structured grid with N_x, N_y and N_z nodes in the x, y and z directions respectively, usual mesh-based discretization strategies imply a challenging issue because the number of nodes involved in the model scales with $N_x \cdot N_y \cdot N_z$, however, by using the separated representation and assuming that the solution involves N modes, one must solve about N 2D problems related to the functions involving the in-plane coordinates (x, y) and the same number of 1D problems related to the functions involving the thickness coordinate z. The computing time related to the solution of the one-dimensional problems can be neglected with respect to the one required for solving the two-dimensional ones. Thus, the resulting complexity scales as $N \cdot N_x \cdot N_y$. By comparing both complexities we can notice that as soon as $N_z \gg N$ the use of separated representations leads to impressive computing time savings, making possible the solution of models never until now solved, and even using light computing platforms.

5.1.4 Numerical results

In this section we consider two laminates with different number of plies, 3 and 29 layers respectively.

5.1.4.1 3-layer laminate

The first laminate of 0.5m $\times$ 0.5m $\times$ 3mm is composed of three plies of similar thicknesses (1mm each), depicted in Fig. 5.1. The in-plane-out-of-plane representation of the electrical field components is expressed from a uniform mesh of the plane domain composed of $50 \times 50 = 2500$ Q1 bilinear finite elements while the thickness is equipped with 1D uniform mesh consisting of 3000 linear elements (1000 elements per layer).

The layers located at the top and bottom are characterized by the electromagnetic properties $\epsilon = 10\epsilon_0$, $\mu = \mu_0$ and $\sigma = 10^{-2}$S/m, with ϵ_0 and μ_0 the vacuum electrical permittivity and the magnetic permeability respectively. The ply located at the center exhibits a larger electrical conductivity in order to attenuate the electrical field, being its associated electromagnetic properties given by $\epsilon = \epsilon_0$, $\mu = \mu_0$ and $\sigma = 10^4$S/m.

We considered such a laminate in order to enforce both kind of electromagnetic losses, the one related to dielectric losses here mostly occurring in the top and bottom layers, whereas in the central layer the losses are motivated by the larger electrical conductivity. The fact of placing the lowest electrical conductivity layers at top and bottom is to ensure the penetration of electromagnetic waves into the laminate to reach the central layer where they will be attenuated due to the larger electrical conductivity. By locating high conductivity layers at the top and bottom the electromagnetic waves cannot reach the central layer because they will be mostly suppressed in the neighborhood of the laminate boundaries.

Dirichlet boundary conditions are enforced on the whole domain boundary, i.e. $\mathbf{E} \times \mathbf{n}$. The tangential components of the electrical field on each face of the layers located on the domain boundary are selected from vector $\mathbf{V}$

$$\mathbf{V} = \begin{pmatrix} \cos kx + \cos ky \\ \cos ky \\ \tau \cos kx \end{pmatrix}, \tag{5.14}$$

with $k = 20\pi$, $\tau = 1$ in the top and bottom layers and with the appropriate value in the central one for ensuring the field jump according to the Gauss law.
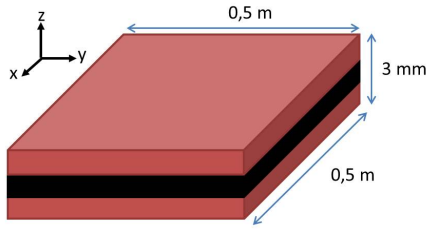
Figure 5.1 – Three-(thin)-plies laminate

.
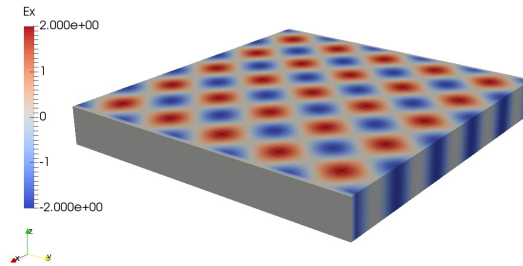
Figure 5.2 – V_x on $\partial\Omega$

.

The components of the electrical field prescribed on the laminate boundary are depicted in Figs. 5.2-5.4 for E_x, E_y and E_z respectively. It is important to remember that only tangential components of the electrical field are enforced on each domain face.

Figs. 5.5-5.7 depict the evolution along the thickness of the real and imaginary parts of the three components of the electric field. In these figures the continuity of E_x and E_y, and the discontinuity of E_z across the ply interfaces can be noticed, with a jump magnitude in agreement with the expected value.

In order to check the effects of unresolved scales related to the penetration depth, we consider a coarser mesh along the thickness composed of 30 elements (10 per layer). It can be noticed that the solution is significantly degraded with respect to solution obtained with the finer mesh as the comparison of Figs. 5.5-5.7 and Figs. 5.8-5.10 reveals, in particular the out-of-plane component of the electric field E_z.
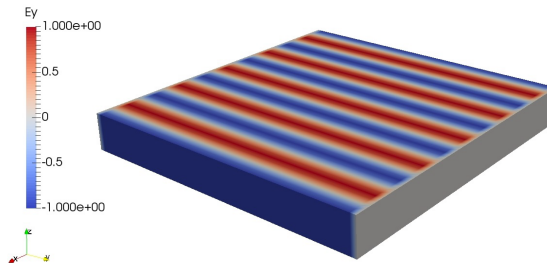
Figure 5.3 – V_y on $\partial\Omega$

.

Figure 5.4 – V_z on $\partial\Omega$

.

Figure 5.5 – Electrical field along the plate thickness, $E_x(0.25, 0.25, z)$

.

Figure 5.6 – Electrical field along the plate thickness, $E_y(0.25, 0.25, z)$

.

Figure 5.7 – Electrical field along the plate thickness, $E_z(0.25, 0.25, z)$

.

Figure 5.8 – Electrical field along the plate thickness, $E_x(0.25, 0.25, z)$ when considering a coarser mesh in the thickness

.
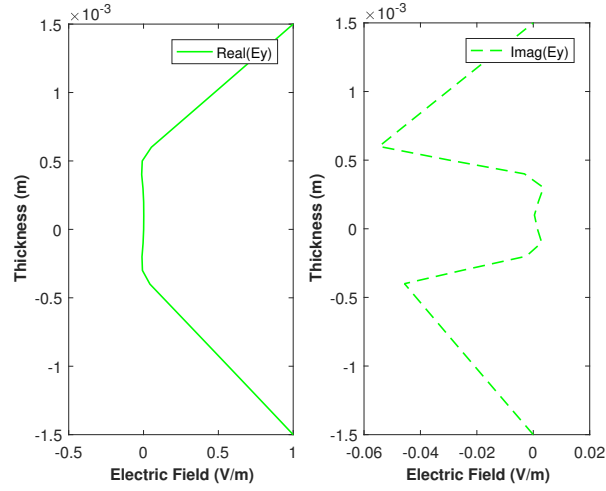
Figure 5.9 – Electrical field along the plate thickness, $E_y(0.25, 0.25, z)$ when considering a coarser mesh in the thickness

.
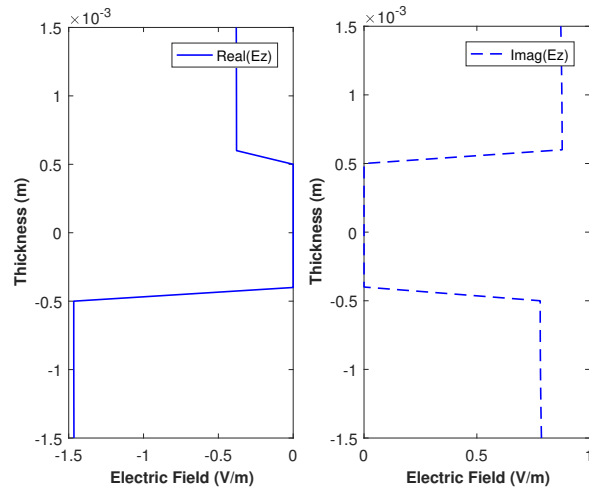
Figure 5.10 – Electrical field along the plate thickness, $E_z(0.25, 0.25, z)$ when considering a coarser mesh in the thickness

.
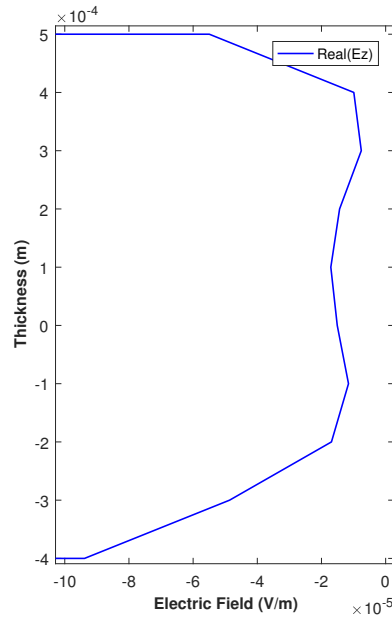
Figure 5.11 – Electrical field along the central layer thickness when considering a mesh in the thickness consisting of 30 elements

.

To better appreciate the unresolved boundary layer (the penetration depth being $\delta = \sqrt{2/\omega\mu_0\sigma} \approx 0.1$mm), Figs. 5.11-5.13 depict the real part of $E_z(0.25, 0.25, z)$ in the central layer.

It can be noticed that for resolving all the scales extremely fine meshes are required along the thickness direction, involving about thousand elements that with the thousands involved in the in-plane resolution imply meshes involving millions of elements (extremely distorted) when proceeding with standard finite elements. When using the in-plane-out-of-plane separated representation, in-plane and out-of-plane meshes become independent avoiding issues related to mesh distortion. On the other hand the problems defined in the thickness are one-dimensional and consequently their computational cost is almost negligible with respect to the solution of in-plane problems. Thus, finally the high-resolution 3D solution is obtained while keeping the computational complexity similar to the one characteristic of the solution of 2D problems (the one defined in the plane).

In order to emphasize the location of both kind of losses, the dielectric one and the one related to eddy currents, Fig. 5.14 represents the density of both losses, where as expected, the dielectric ones locate at the external layers, whereas the one related to eddy current effects locates in the central layer and close to the interfaces where the electric field is attenuated (the penetration deept).

5.1.4.2 29-layer laminate

The second laminate of 0.5m $\times 0.5$m $\times 2.9$mm is composed of 29 plies of similar thicknesses (0.1mm each) where two different materials are alternatively placed as illustrated in Fig. 5.15. The in-plane-out-of-plane representation of the electrical field components is expressed
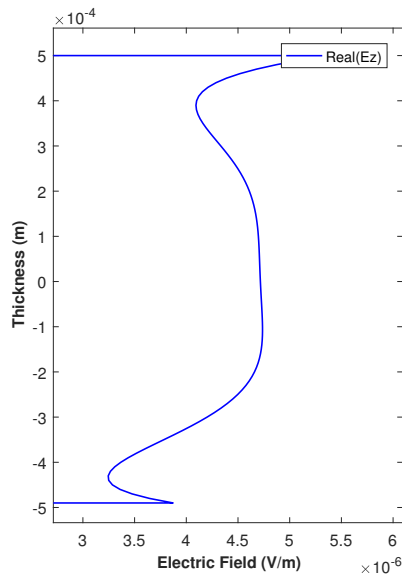
Figure 5.12 – Electrical field along the central layer thickness when considering a mesh in the thickness consisting of 300 elements

.
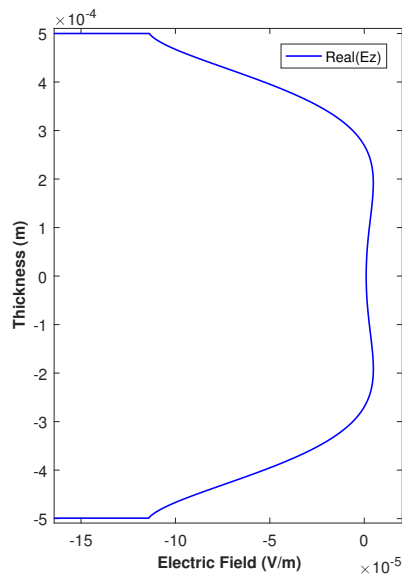
Figure 5.13 – Electrical field along the central layer thickness when considering a mesh in the thickness consisting of 3000 elements

.
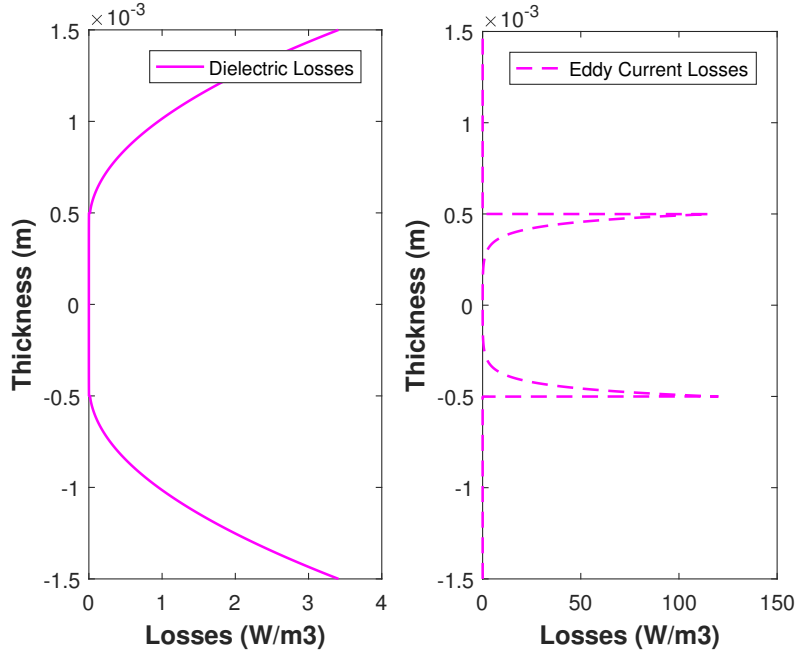
Figure 5.14 – Electromagnetic losses: (left) dielectric and (right) eddy current
.

from a uniform mesh of the plane domain composed of $50 \times 50 = 2500$ Q1 bilinear finite
elements while the thickness is equipped with 1D uniform mesh consisting of 1450 linear
elements (50 elements per layer).

The layers in red in Fig.   5.15 are characterized by the electromagnetic properties
$\epsilon = 5\epsilon_0$, $\mu = \mu_0$ and $\sigma = 0$S/m, with ϵ_0 and μ_0 the vacuum electrical permittivity and
the magnetic permeability respectively. The remaining plies exhibit a larger electrical con-
ductivity in order to attenuate the electrical field, being its associated electromagnetic
properties given by $\epsilon = \epsilon_0$, $\mu = \mu_0$ and $\sigma = 1$S/m. The boundary conditions are the same
that were enforced when addressing the solution of the three-layer laminate.

Figs. 5.16-5.18 depict the evolution along the thickness of the real and imaginary parts
of the three components of the electric field.   In these figures the continuity of E_x and
E_y, and the discontinuity of E_z across the ply interfaces can be noticed, with a jump
magnitude in agreement with the expected value.   It is important to note that such a
high-resolution remains out-of-reach when using standard 3D mesh-based discretization
techniques, whereas the separated representation allows a very accurate solution, cheap
and fast from the computational view point (the solution was obtained in 20 seconds using
a standard laptop).

5.1.5   Discussion and conclusions

The Proper Generalized Decomposition within the In-Plane-Out-Of-Plane separation has
been succesfully applied to address the propagation of electromagnetic waves in compos-
ite laminates. Because its inherent throughout-its-thickness high-resolution, direct conse-
quence of the in-plane-out-of-plane separated representation, the reached resolution falls
beyond the capabilities of standard and well experienced mesh-based discretization tech-
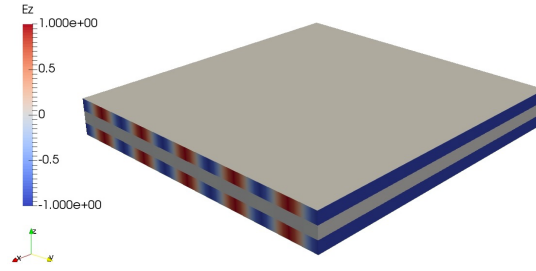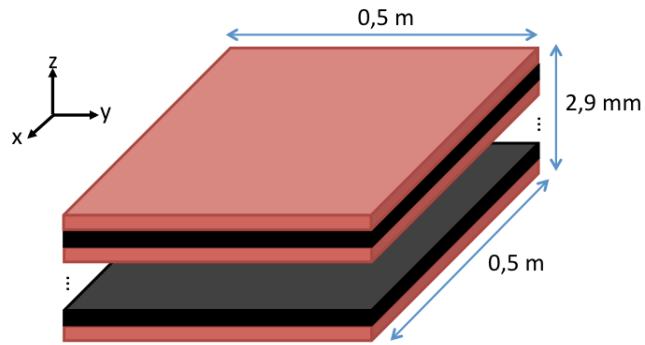
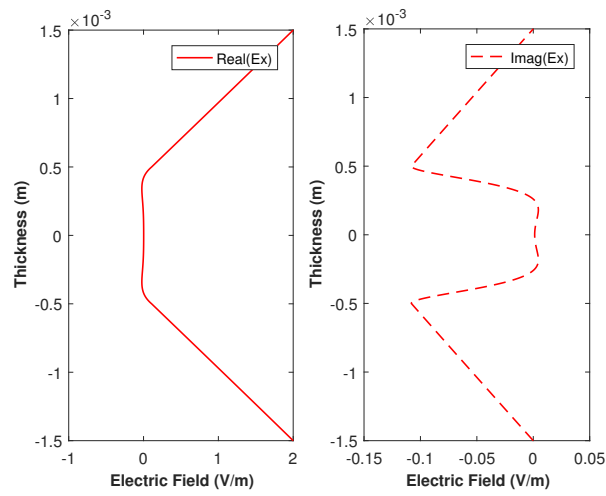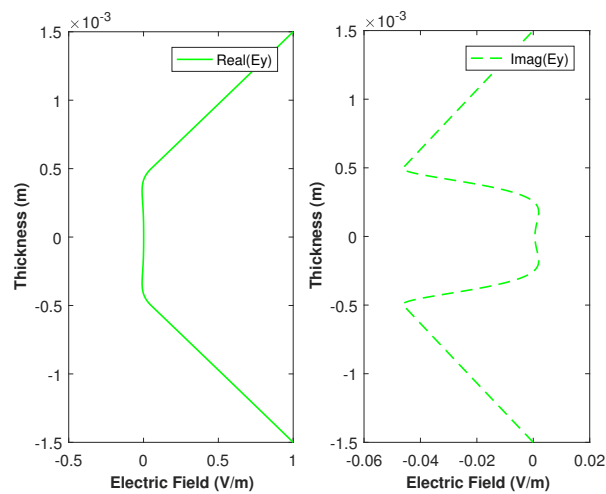Figure 5.15 – 29-plies laminate

.
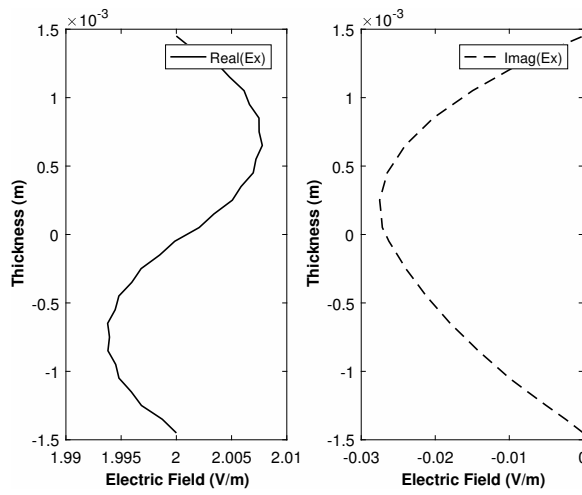
Figure 5.16 – Electrical field along the plate thickness, $E_x(0.25, 0.25, z)$

.

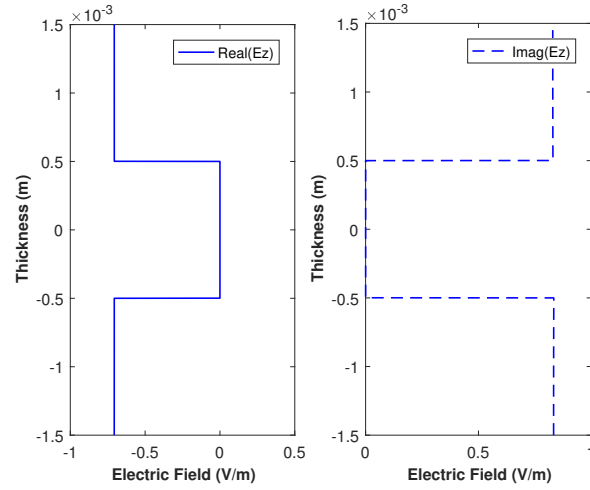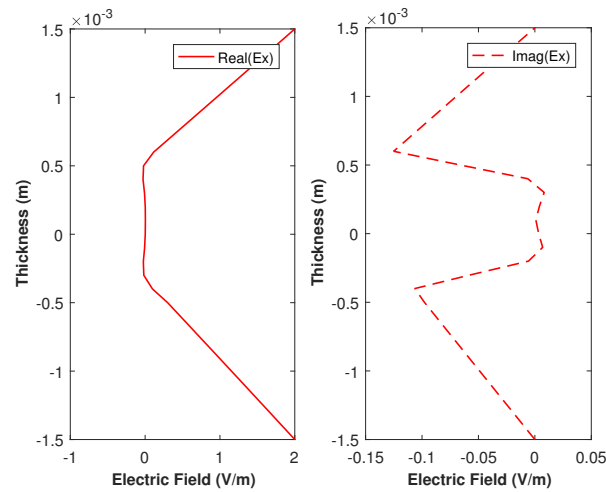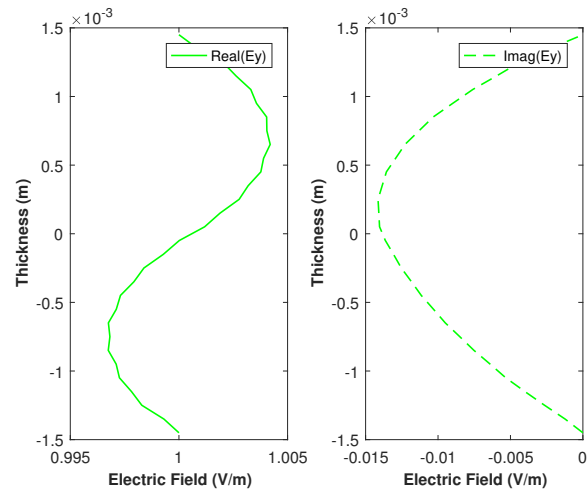Figure 5.17 – Electrical field along the plate thickness, $E_y(0.25, 0.25, z)$

.
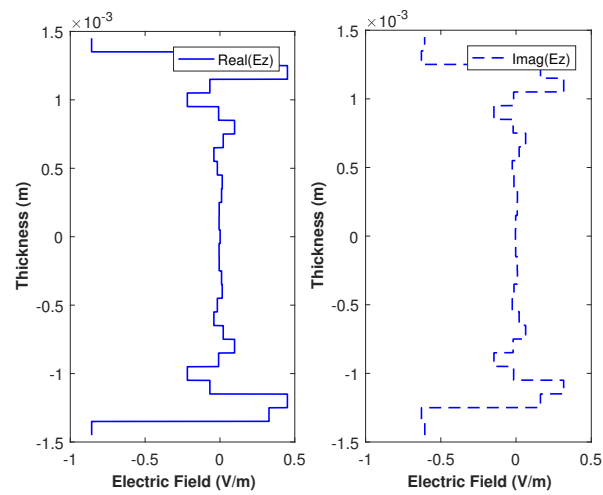
Figure 5.18 – Electrical field along the plate thickness, $E_z(0.25, 0.25, z)$

.

niques. By using it, micrometric resolutions can be easily reached when addressing usual composite structural parts. It allows zooming at regions exhibiting high variations of the solution, as for example the wave attenuation when reaching a conductive layer, that localizes the solution in an extremely narrow layer, whose accurate representation remains out of reach for standard discretization techniques.

The second advantage in uncoupling plane coordinates from the thickness coordinate is related to the facility to enforce the jump of the normal component of the electrical field across two layers exhibiting different electromagnetic properties. For that purpose the field approximation is able to produce jumps at the laminate interfaces.

When using a high-resolution discretization based on the in-plane-out-of-plane separated representation, the double-curl electromagnetic formulation worked quite well as soon as boundary conditions were regular enough. However as soon as complex Dirichlet boundary conditions were enforced a lack of convergence was noticed and the computed solutions were almost wrong. In fact the Gauss law was not fulfilled at the discrete level and for that reason a regularized formulation was considered, that allowed ensuring the solution convergence.

These three issues are subtly entangled. It is important to note that when solving the electromagnetic problem in the laminate, the z-component of the electrical field E_z is not enforced on the top and bottom surfaces. Moreover, at the ply interfaces E_z jumps according to the Gauss law, and then, if the field attenuation when the wave reaches the central (conductive) layer is not accurately described (approximated), the solution E_z becomes wrong almost everywhere.

The consideration of perfect conductors as well as the domain decomposition for coupling standard discretization techniques and PGD separated representation, outside and inside the composite laminate respectively, are the main works in progress.

5.2 Explicit-Implicit Hybrid Time Marching Scheme for Thin Geometries

This work presents a novel explicit-implicit time marching scheme suitable for thin geometries. The proposed methology alleviates unaffordable time steps arising from the thinnest dimension while keeping the advantages of explicit time integration schemes. This section is structed as follows: Subsection 5.2.1 defines the elastodynamics framework and the proposed in-plane-out-of-plane separated representation. Subsection 5.2.2 revisits classical time integration within the separated representation framework, and proposes an efficient hybrid explicit/implicite formulation. Finally, Subsection 5.2.3 validates the proposed methodology from some case studies.

5.2.1   Elastodynamics: Problem definition

We consider a physical domain Ω for which a linear elastic behavior is assumed, according to

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\epsilon}, \qquad (5.15)$$

where $\mathbf{C}$ is the fourth order stiffness tensor, and the strain tensor $\boldsymbol{\epsilon}$ derives from the symmetric component of the gradient of displacements i.e. $\boldsymbol{\epsilon} = \nabla_s \mathbf{u}$, where ∇_s refers to the symmetric component.

From now on we consider Voigt notation, and for the sake of notational simplicity we consider the same notation, $\boldsymbol{\sigma}$, $\boldsymbol{\epsilon}$ and $\mathbf{C}$ for expressing the stress and strain vectors and the

stiffness matrix respectively.

The dynamic problem, in absence of damping and external forces, with the displacement field $\mathbf{u}(\mathbf{x}, t)$ for $\mathbf{x} \in \Omega$ and $t \in I = [0, T]$, reads

$$\rho \ddot{\mathbf{u}}(\mathbf{x}, t) = \nabla \cdot \boldsymbol{\sigma}, \tag{5.16}$$

with ρ the material density, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ the first and second time derivative of the displacement field respectively, i.e. the velocity and acceleration.

The domain boundary $\Gamma = \partial \Omega$ is partitioned in the so-called Dirichlet and Neumann regions, Γ_D and Γ_N, where respectively displacements and tractions are enforced, with $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Dynamic problems require specifying the initial displacement and velocity that without loss of generality in what follows are assumed null, i.e. $\dot{\mathbf{u}}(\mathbf{x}, t = 0) = \mathbf{0}$ and $\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{0}$.

Assuming again the trial and test displacements belonging to appropriate functional spaces, and considering an elastic constitutive equation, the weak form associated with (5.16) reads

$$\rho \int_{\Omega} \mathbf{u}^* \cdot \ddot{\mathbf{u}} \, d\mathbf{x} + \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot (\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u})) \, d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{F} \, d\mathbf{x}, \tag{5.17}$$

where the applied traction depends on time, i.e. $\mathbf{F} = \mathbf{F}(t)$.

5.2.1.1   In-plane-out-of-plane separated representation

As discussed in the previous section, with Ω having one dimension (the one related to the thickness) much smaller than the others involving the in-plane coordinates, an in-plane-out-of-plane separated representation seems again the most appealing route for addressing 3D discretizations while keeping the computational complexity the one characteristic of 2D discretizations. The domain is expressed from $\Omega = \Omega_{xy} \times \Omega_z$.

Even if as also indicated space-time separated discretizations were considered many times in the past [Ladeveze 1985, Ammar *et al.* 2007], in the present work time derivatives are discretized using standard schemes.

By considering the notation $\mathbf{u}(x, y, z, t = k\Delta t) = \mathbf{u}^k(x, y, z)$, with Δt the time step, the in-plane-out-of-plane separated representation of the displacement field at time $t_k = k\Delta t$, $\mathbf{u}^k(x, y, z)$, reads

$$\mathbf{u}^k(x, y, z) = \begin{pmatrix} u^k(x, y, z) \\ v^k(x, y, z) \\ w^k(x, y, z) \end{pmatrix} \approx \mathbf{u}_N^k(x, y, z) = \sum_{i=1}^{N} \begin{pmatrix} u_{xy}^{i,k}(x, y) \cdot u_z^{i,k}(z) \\ v_{xy}^{i,k}(x, y) \cdot v_z^{i,k}(z) \\ w_{xy}^{i,k}(x, y) \cdot w_z^{i,k}(z) \end{pmatrix} =$$

$$\sum_{i=1}^{N} \mathbf{U}_{xy}^{i,k}(x, y) \circ \mathbf{U}_z^{i,k}(z) \tag{5.18}$$

where "$\circ$" refers to the Hadamard product, and with

$$\mathbf{U}_{xy}^{i,k}(x, y) = \begin{pmatrix} u_{xy}^{i,k}(x, y) \\ v_{xy}^{i,k}(x, y) \\ w_{xy}^{i,k}(x, y) \end{pmatrix} = \begin{pmatrix} u_{xy}^{i,k} \\ v_{xy}^{i,k} \\ w_{xy}^{i,k} \end{pmatrix}, \tag{5.19}$$

$$\mathbf{U}_z^{i,k}(z) = \begin{pmatrix} u_z^{i,k}(z) \\ v_z^{i,k}(z) \\ w_z^{i,k}(z) \end{pmatrix} = \begin{pmatrix} u_z^{i,k} \\ v_z^{i,k} \\ w_z^{i,k} \end{pmatrix}, \tag{5.20}$$

where for alleviating the notation the coordinate dependences will be omitted.

From all them we can obtain the separated vector form of the strain tensor at time t_k, $\boldsymbol{\epsilon}^k \equiv \boldsymbol{\epsilon}(\mathbf{u}^k)$:

$$\boldsymbol{\epsilon}(\mathbf{u}^k) \approx \sum_{i=1}^{N} \begin{pmatrix} \frac{\partial u_{xy}^{i,k}}{\partial x} \cdot u_z^{i,k} \\ \frac{\partial v_{xy}^{i,k}}{\partial y} \cdot v_z^{i,k} \\ w_{xy}^{i,k} \cdot \frac{\partial w_z^{i,k}}{\partial z} \\ \frac{\partial u_{xy}^{i,k}}{\partial y} \cdot u_z^{i,k} + \frac{\partial v_{xy}^{i,k}}{\partial x} \cdot v_z^{i,k} \\ \frac{\partial w_{xy}^{i,k}}{\partial x} \cdot w_z^{i,k} + u_{xy}^{i,k} \cdot \frac{\partial u_z^{i,k}}{\partial z} \\ \frac{\partial w_{xy}^{i,k}}{\partial y} \cdot w_z^{i,k} + v_{xy}^{i,k} \cdot \frac{\partial v_z^{i,k}}{\partial z} \end{pmatrix} . \tag{5.21}$$

The separated representation constructor proceeds by computing a term of the sum at each iteration. Assuming that the first $n-1$ modes (terms of the finite sum) of the solution were already computed, $\mathbf{u}_{n-1}^k(x, y, z)$ with $n \geq 1$, the solution enrichment reads:

$$\mathbf{u}_n^k(x, y, z) = \mathbf{u}_{n-1}^k(x, y, z) + \mathbf{U}_{xy}^{n,k}(x, y) \circ \mathbf{U}_z^{n,k}(z) \tag{5.22}$$

where both vectors $\mathbf{U}_{xy}^{n,k}$ and $\mathbf{U}_z^{n,k}$ are unknown at the present iteration defining a nonlinear problem. The test function $\mathbf{u}^*$ reads $\mathbf{u}^* = \mathbf{U}_{xy}^* \circ \mathbf{U}_z^{n,k} + \mathbf{U}_{xy}^{n,k} \circ \mathbf{U}_z^*$.

With both $\mathbf{U}_{xy}^{n,k}$ and $\mathbf{U}_z^{n,k}$ unknown the resulting problem becomes non-linear. We proceed by considering the simplest linearization strategy, an alternated directions fixed point algorithm widely considered and described in our former works.

When $\mathbf{U}_z^{n,k}$ is assumed known, we consider the test function $\mathbf{u}^\star$ given by $\mathbf{U}_{xy}^\star \circ \mathbf{U}_z^{n,k}$. By introducing the trial and test functions into the weak form and then integrating in Ω_z because all the functions depending on the thickness coordinate are known, we obtain a 2D weak formulation defined in Ω_{xy} whose discretization (by using a standard discretization strategy, e.g. finite elements) allows computing $\mathbf{U}_{xy}^{n,k}$.

Analogously, when $\mathbf{U}_{xy}^{n,k}$ is assumed known, the test function $\mathbf{u}^\star$ is given by $\mathbf{U}_{xy}^{n,k} \circ \mathbf{U}_z^\star$. By introducing the trial and test functions into the weak form and then integrating in Ω_{xy} because all the functions depending on the in-plane coordinates (x, y) are at present known, we obtain a 1D weak formulation defined in Ω_z whose discretization (using any technique for solving standard ODE equations) allows computing $\mathbf{U}_z^{n,k}$.

Thus, the 3D computational cost is transformed into a sequence of 2D and 1D solutions, with the associated computing time savings [Bognet *et al.* 2012].

5.2.2 Time discretization

Before introducing the hybrid strategy we consider at time t_{k+1} the standard implicit and explicit formulations (two commun time integration schemas among other possibilities), given respectively by

$$\rho \int_\Omega \mathbf{u}^* \cdot \frac{\mathbf{u}^{k+1} - 2\mathbf{u}^k + \mathbf{u}^{k-1}}{\Delta t^2} d\mathbf{x} + \int_\Omega \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot \left(\mathbf{C}\boldsymbol{\epsilon}\left(\frac{\mathbf{u}^{k+1} + \mathbf{u}^{k-1}}{2} \right) \right) d\mathbf{x} =$$

$$\int_{\Gamma_N} \mathbf{u}^* \cdot \frac{\mathbf{F}^{k+1} + \mathbf{F}^{k-1}}{2} d\mathbf{x}, \tag{5.23}$$

that as previously indicated is unconditionally stable, and the explicit one

$$\rho \int_\Omega \mathbf{u}^* \cdot \frac{\mathbf{u}^{k+1} - 2\mathbf{u}^k + \mathbf{u}^{k-1}}{\Delta t^2} d\mathbf{x} + \int_\Omega \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot \left(\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u}^k) \right) d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{F}^k d\mathbf{x}, \tag{5.24}$$

that is conditionally stable, with the stability limit Δt_{max}, defining the stability domain $\Delta t < \Delta t_{max}$, given by

$$\Delta t_{max} = \frac{L}{c}, \tag{5.25}$$

where L is the characteristic length of the spatial discretization and the dilatational wave speed c is given by

$$c = \sqrt{\frac{E(1-\nu)}{(1+\nu)(1-2\nu)\rho}}. \tag{5.26}$$

As previously commented explicit strategies are employed in many commercial codes. However, when applied to discretize 3D problems defined in degenerated domains, like plates or shells, the extremely fine meshes considered along the thickness direction have an unfavorable impact on the time step that becomes extremely small to ensure stability. The in-plane-out-of-plane separated representation cannot scape to this important issue, being the mesh size along the out-of-plane coordinate (much finer that the one used in the plane) the one that determines the time step.

It is important emphasizing the main aim of the present work and the proposed methodology for performing it. First, it is important to note that we are interested in performing fully 3D simulations in degenerated geometries (e.g. plate domains) while retaining as many explicit time integration features as possible.

In this context the following remarks can be addressed:

- When using 2D discrete models (considering for example plate elements), the stability criterion related to explicit time integrations involves the size of the elements, but as the mesh is the one related to the middle plane, the critical time step remains reasonable in most of cases;

- However, as soon as 3D discretizations are considered, the characteristic size of the finite elements along the plate thickness becomes much smaller than the in-plane characteristic length, and then when considering explicit time integrations the time step needed for ensuring stability decreases with the through-of-thickness characteristic element length;

- Increasing the resolution in the thickness direction implies the increase of the number of elements involved in the discretization as well as the decrease of the time step for ensuring stability, both having unfavorable consequences on the computational cost;

- In former works [Bognet et al. 2012, Bognet et al. 2014] we proposed in the framework of elastostatics considering in-plane-out-of-plane separated representations that allowed reducing the computational complexity of solving a fully 3D problem to the one characteristic of 2D solutions;

- However, as just indicated, such a decomposition when combined with explicit time integrations fails, because again the stability is associated to the smallest discretization characteristic length, the one related to the through-of-thickness discretization;

- It is in that impasse that one is tempted of using, in the case of explicit time integration, the in-plane-out-of-plane separated representation (that reduces the computational complexity to the one characteristic of 2D models) combined with an hybrid time integration, explicit in the plane (conditionally stable but with the critical time-step scaling with the characteristic in-plane discretization length) and implicit along the thickness (unconditionally stable), that allows both reducing the computational complexity while keeping as stability constraint the one associated to the in-plane explicit time integration;

- Obviously fully implicit in-plane-out-of-plane decompositions are possible, where the implicit time integration ensures unconditional stability while the space separated representation reduces the computational complexity. Despite of its intrinsic interest it is not considered in the present paper, and in all cases, the associated solutions are the same as the ones obtained by using a fully 3D finite element discretization but reducing the solution computational complexity. As previously commented fully explicit integrations fail because the too stringent stability conditions induced by the too fine through-of-thickness discretization;

Thus, in this work we analyze an intermediate procedure, the one in which the fine through-of-thickness representation is alleviated thanks to the use of the in-plane-out-of-plane space separated representation together with an implicit time integration in all operators involving derivatives throughout the thickness coordinate. Thus, the stability of the resulting discretization is expected being induced by the in-plane mesh in which an explicit time integration is retained. The present work is intended analyzing this hybrid methodology, and proving that in the case of fully explicit separated representations (as in the case of fully explicit 3D finite elements) the stability is dictated by the smallest characteristic discretization length (the one along the domain thickness). On the contrary when considering the hybrid scheme described herein, we expect the stability being dictated by the characteristic in-plane discretization length (being the through-of-thinness discretization implicit). In summary, the main goal is enriching explicit 2D plate and shell formulations widely employed in industry and commercial codes, with a fine through-of-thickness description (3D) without affecting unfavorably the integration stability.

5.2.2.1 Explicit-in-plane / implicit-out-of-plane hybrid scheme

As just indicated, in order to circumvent the just referred stability issues, we propose an out-of-plane implicit discretization (unconditionally stable) while the in-plane discretization (implying coarser meshes) makes use of an explicit schema. Thus, the stability is precribed by the in-plane size mesh, several order of magnitude higher than the one associated to the thickness.

For that purpose we propose considering at time t_k the strain defined by

$$
\boldsymbol{\epsilon}^h(\mathbf{u}^k) = \begin{pmatrix} u^k_{,x} \\ v^k_{,y} \\ \frac{w^{k+1}_{,z}+w^{k-1}_{,z}}{2} \\ \frac{v^{k+1}_{,z}+v^{k-1}_{,z}}{2} + w^k_{,y} \\ \frac{u^{k+1}_{,z}+u^{k-1}_{,z}}{2} + w^k_{,x} \\ u^k_{,y} + v^k_{,x} \end{pmatrix} \approx
$$

$$
\begin{pmatrix} \sum_{i=1}^{N_k} u^{i,k}_{xy,x} \cdot u^{i,k}_z \\ \sum_{i=1}^{N_k} v^{i,k}_{xy,y} \cdot v^{i,k}_z \\ \frac{\sum_{i=1}^{N_{k+1}} w^{i,k+1}_{xy} \cdot w^{i,k+1}_{z,z} + \sum_{i=1}^{N_{k-1}} w^{i,k-1}_{xy} \cdot w^{i,k-1}_{z,z}}{2} \\ \frac{\sum_{i=1}^{N_{k+1}} v^{i,k+1}_{xy} \cdot v^{i,k+1}_{z,z} + \sum_{i=1}^{N_{k-1}} v^{i,k-1}_{xy} \cdot v^{i,k-1}_{z,z}}{2} + \sum_{i=1}^{N_k} w^{i,k}_{xy,y} \cdot w^{i,k}_z \\ \frac{\sum_{i=1}^{N_{k+1}} u^{i,k+1}_{xy} \cdot u^{i,k+1}_{z,z} + \sum_{i=1}^{N_{k-1}} u^{i,k-1}_{xy} \cdot u^{i,k-1}_{z,z}}{2} + \sum_{i=1}^{N_k} w^{i,k}_{xy,x} \cdot w^{i,k}_z \\ \sum_{i=1}^{N_k} u^{i,k}_{xy,y} \cdot u^{i,k}_z + \sum_{i=1}^{N_k} v^{i,k}_{xy,x} \cdot v^{i,k}_z \end{pmatrix} \tag{5.27}
$$

where for the sake of notational simplicity the derivatives of function $u_\bullet$ with respect the coordinate x is noted by $u_{\bullet,x}$ (and similarly for the other functions involved in the displacement components with respect to any coordinate). Moreover, the superscript $\bullet^h$ refers to

its hybrid nature and N_{k+1}, N_k, N_{k-1} are the number of products involved in the separated representation of the displacement at times steps $k+1, k$ and $k-1$ respectively.

It can be noticed that the derivatives involving the out-of-plane (thickness) coordinate are treated using an implicit schema whereas an explicit one is retained for the in-plane derivatives. Thus, the hybrid schema is some place in between standard implicit and explicit techniques, taking profit of the advantages of both them.

When using the hybrid schema the weak form at time t_{k+1}, consists of finding $\mathbf{u}^{k+1}$, verifying

$$\int_{\Omega} \rho \mathbf{u}^* \cdot \frac{\mathbf{u}^{k+1} - 2\mathbf{u}^k + \mathbf{u}^{k-1}}{\Delta t^2} d\mathbf{x} + \int_{\Omega} \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot \left(\mathbf{C}\boldsymbol{\epsilon}^h(\mathbf{u})\right) d\mathbf{x} = \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{F}^k d\mathbf{x}. \qquad (5.28)$$

To construct the separated representation of the solution at time t_{k+1} we consider the standard procedure, assuming that $n-1$ terms have been already computed and that at the present iteration looks for the term n, according to

$$\mathbf{u}_{n-1}^{k+1}(x,y,z) = \begin{pmatrix} u_{n-1}^{k+1}(x,y,z) \\ v_{n-1}^{k+1}(x,y,z) \\ w_{n-1}^{k+1}(x,y,z) \end{pmatrix} = \sum_{i=1}^{n-1} \mathbf{U}_{xy}^{i,k+1}(x,y) \circ \mathbf{U}_z^{i,k+1}(z), \qquad (5.29)$$

with

$$\mathbf{u}_n^{k+1}(x,y,z) = \mathbf{u}_{n-1}^{k+1}(x,y,z) + \mathbf{P}^{k+1}(x,y) \circ \mathbf{T}^{k+1}(z). \qquad (5.30)$$

where for the sake of notational simplicity the unknown fields $\mathbf{U}_{xy}^{n,k+1}$ and $\mathbf{U}_z^{n,k+1}$ are referred by $\mathbf{P}^{k+1}(x,y)$ and $\mathbf{T}^{k+1}(z)$, with components

$$\mathbf{P}^{k+1}(x,y) = \begin{pmatrix} p_u^{k+1}(x,y) \\ p_v^{k+1}(x,y) \\ p_w^{k+1}(x,y) \end{pmatrix}, \qquad (5.31)$$

and

$$\mathbf{T}^{k+1}(z) = \begin{pmatrix} t_u^{k+1}(z) \\ t_v^{k+1}(z) \\ t_w^{k+1}(z) \end{pmatrix}. \qquad (5.32)$$

The linearity allows writing

$$\boldsymbol{\epsilon}_n^h(x,y,z) = \boldsymbol{\epsilon}_{n-1}^h(x,y,z) + \boldsymbol{\epsilon}^{\boldsymbol{h}}{}_{PT}(x,y,z) \qquad (5.33)$$

where

$$\boldsymbol{\epsilon}_{n-1}^h(x,y,z) = \begin{pmatrix} u_{,x}^k \\ v_{,y}^k \\ \frac{w_{n-1,z}^{k+1} + w_{,z}^{k-1}}{2} \\ \frac{v_{n-1,z}^{k+1} + v_{,z}^{k-1}}{2} + w_{,y}^k \\ \frac{u_{n-1,z}^{k+1} + u_{,z}^{k-1}}{2} + w_{,x}^k \\ u_{,y}^k + v_{,x}^k \end{pmatrix} =$$

$$\begin{pmatrix} \sum_{i=1}^{N_k} u_{xy,x}^{i,k} \cdot u_z^{i,k} \\ \sum_{i=1}^{N_k} v_{xy,y}^{i,k} \cdot v_z^{i,k} \\ \frac{\sum_{i=1}^{n-1} w_{xy}^{i,k+1} \cdot w_{z,z}^{i,k+1} + \sum_{i=1}^{N_{k-1}} w_{xy}^{i,k-1} \cdot w_{z,z}^{i,k-1}}{2} \\ \frac{\sum_{i=1}^{n-1} v_{xy}^{i,k+1} \cdot v_{z,z}^{i,k+1} + \sum_{i=1}^{N_{k-1}} v_{xy}^{i,k-1} \cdot v_{z,z}^{i,k-1}}{2} + \sum_{i=1}^{N_k} w_{xy,y}^{i,k} \cdot w_z^{i,k} \\ \frac{\sum_{i=1}^{n-1} u_{xy}^{i,k+1} \cdot u_{z,z}^{i,k+1} + \sum_{i=1}^{N_{k-1}} u_{xy}^{i,k-1} \cdot u_{z,z}^{i,k-1}}{2} + \sum_{i=1}^{N_k} w_{xy,x}^{i,k} \cdot w_z^{i,k} \\ \sum_{i=1}^{N_k} u_{xy,y}^{i,k} \cdot u_z^{i,k} + \sum_{i=1}^{N_k} v_{xy,x}^{i,k} \cdot v_z^{i,k} \end{pmatrix}, \qquad (5.34)$$

and

$$\boldsymbol{\epsilon}_{PT}^h(x, y, z) = \begin{pmatrix} 0 \\ 0 \\ \frac{p_w^{k+1} \cdot t_{w,z}^{k+1}}{2} \\ \frac{p_v^{k+1} \cdot t_{v,z}^{k+1}}{2} \\ \frac{p_u^{k+1} \cdot t_{u,z}^{k+1}}{2} \\ 0 \end{pmatrix}.$$

The test displacement reads

$$\mathbf{u}^*(x, y, z) = \begin{pmatrix} p_u^*(x,y) \cdot t_u^{k+1}(z) + p_u^{k+1}(x,y) \cdot t_u^*(z) \\ p_v^*(x,y) \cdot t_v^{k+1}(z) + p_v^{k+1}(x,y) \cdot t_v^*(z) \\ p_w^*(x,y) \cdot t_w^{k+1}(z) + p_w^{k+1}(x,y) \cdot t_w^*(z) \end{pmatrix} =$$

$$\mathbf{P}^* \circ \mathbf{T}^{k+1} + \mathbf{P}^{k+1} \circ \mathbf{T}^*, \tag{5.35}$$

and the associated strain

$$\boldsymbol{\epsilon}(\mathbf{u}^*(x, y, z)) = \begin{pmatrix} p_{u,x}^* \cdot t_u^{k+1} + p_{u,x}^{k+1} \cdot t_u^* \\ p_{v,y}^* \cdot t_v^{k+1} + p_{v,y}^{k+1} \cdot t_v^* \\ p_w^* \cdot t_{w,z}^{k+1} + p_w^{k+1} \cdot t_{w,z}^* \\ p_{w,y}^* \cdot t_w^{k+1} + p_{w,y}^{k+1} \cdot t_w^* + p_v^* \cdot t_{v,z}^{k+1} + p_v^{k+1} \cdot t_{v,z}^* \\ p_{w,x}^* \cdot t_w^{k+1} + p_{w,x}^{k+1} \cdot t_w^* + p_u^* \cdot t_{u,z}^{k+1} + p_u^{k+1} \cdot t_{u,z}^* \\ p_{v,x}^* \cdot t_v^{k+1} + p_{v,x}^{k+1} \cdot t_v^* + p_{u,y}^* \cdot t_u^{k+1} + p_{u,y}^{k+1} \cdot t_u^* \end{pmatrix}. \tag{5.36}$$

For the sake of simplicity, and without loss of generality, we assume that the applied traction $\mathbf{F}$ on Γ_N, can be expressed from the single term separated representation, i.e.

$$\mathbf{F}^k(x, y, z) = \mathbf{F}_{x,y}^k(x, y) \circ \mathbf{F}_z^k(z). \tag{5.37}$$

Using the previous expressions, Eq. (5.28) reads

$$\int_\Omega \rho \mathbf{u}^* \cdot \frac{\mathbf{P}^{k+1} \circ \mathbf{T}^{k+1}}{\Delta t^2} \, d\mathbf{x} + \int_\Omega \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot \left(\mathbf{C} \boldsymbol{\epsilon}_{PT}^h \right) \, d\mathbf{x} =$$

$$-\int_\Omega \rho \mathbf{u}^* \cdot \frac{\mathbf{u}_{n-1}^{k+1} - 2\mathbf{u}^k + \mathbf{u}^{k-1}}{\Delta t^2} \, d\mathbf{x} - \int_\Omega \boldsymbol{\epsilon}(\mathbf{u}^*) \cdot \left(\mathbf{C} \boldsymbol{\epsilon}_{n-1}^h \right) \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{F}^k \, d\mathbf{x}. \tag{5.38}$$

As both $\mathbf{P}^{k+1}$ and $\mathbf{T}^{k+1}$ are unknown, problem (5.38) becomes nonlinear and consequently requires an appropriate linearization strategy. As as usual in our previous works an alternated directions fixed point strategy is considered that by assuming $\mathbf{T}^{k+1}$ known calculates $\mathbf{P}^{k+1}$ and from the last updates $\mathbf{T}^{k+1}$. The process continues until reaching convergence (the fixed point).

When assuming $\mathbf{T}^{k+1}$ known the test displacement reads

$$\mathbf{u}^*(x, y, z) = \begin{pmatrix} p_u^*(x,y) \cdot t_u^{k+1}(z) \\ p_v^*(x,y) \cdot t_v^{k+1}(z) \\ p_w^*(x,y) \cdot t_w^{k+1}(z) \end{pmatrix} = \mathbf{P}^* \circ \mathbf{T}^{k+1}, \tag{5.39}$$

that introduced into the weak form (5.38) results in a 2D problem involving the in-plane coordinates that allows calculating $\mathbf{P}^{k+1}$. Now, assuming that last known, the test displacement becomes

$$\mathbf{u}^*(x, y, z) = \begin{pmatrix} p_u^{k+1}(x,y) \cdot t_u^*(z) \\ p_v^{k+1}(x,y) \cdot t_v^*(z) \\ p_w^{k+1}(x,y) \cdot t_w^*(z) \end{pmatrix} = \mathbf{P}^{k+1} \circ \mathbf{T}^*, \tag{5.40}$$
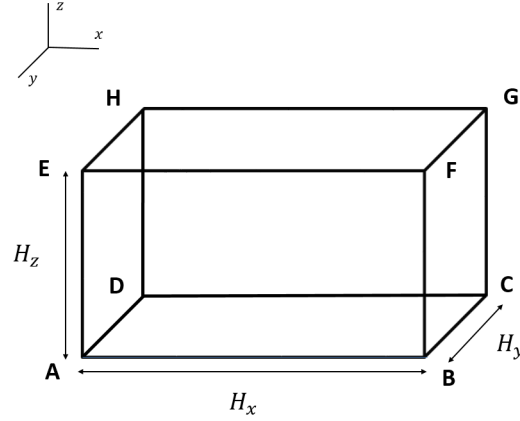
Figure 5.19 – Problem geometry

that introduced at its turn into the weak form (5.38) results in a 1D problem involving the thickness, whose solution results in $\mathbf{T}^{k+1}$. As preciously indicated the alternate direction procedure continues until reaching the convergence or stagnation criteria characterized by

$$\frac{\int_\Omega \left(\mathbf{P}_p^{k+1} \circ \mathbf{T}_p^{k+1} - \mathbf{P}_{p-1}^{k+1} \circ \mathbf{T}_{p-1}^{k+1}\right)^2 d\mathbf{x}}{\int_\Omega \left(\mathbf{P}_{p-1}^{k+1} \circ \mathbf{T}_{p-1}^{k+1}\right)^2 d\mathbf{x}} < \epsilon, \tag{5.41}$$

where p refers to the fixed point iteration and ϵ is a small enough threshold value.

Similarly the enrichment procedure stops when the $L2$ norm of the new enrichment

$$\frac{\int_\Omega \left(\mathbf{P}^{k+1} \circ \mathbf{T}^{k+1}\right)^2 d\mathbf{x}}{\int_\Omega \left(\mathbf{U}_{xy}^{1,k+1} \circ \mathbf{U}_z^{1,k+1}\right)^2 d\mathbf{x}} < \tilde{\epsilon}, \tag{5.42}$$

is fulfilled, with $\tilde{\epsilon}$ another small enough threshold value.

5.2.3 Numerical validation

5.2.3.1 Dynamics of an homogeneous plate

We consider the problem defined in the domain Ω depicted in Fig. 5.19, with $H_x = H_y = 3m$ and $H_z = 0.1m$. In the first case study, the material occupying Ω is assumed isotropic.

Boundary conditions are given by: $\mathbf{u} = (0, 0, 0)$ on the face ADHE; $\mathbf{u} = (free, 0, free)$ on the faces ABFE and DCGH; $\mathbf{F(t)} = (0, 0, A\sin(\omega t))$ on face BCGF, with $A = 10^8$, $\omega = 20\pi$ as depicted in Fig. 5.20.

The material properties are defined in Table 5.1, where E is the Young modulus, ν the Poisson coefficient and ρ the material density.

Figures 5.21 and 5.22 compare the stability of standard explicit Q8-3D finite elements (fully explicit separated representations with equivalent discretizations lead to the same results) and the hybrid scheme just proposed, for different values of the in-plane and out-of-plane mesh sizes, L_{xy} and L_z respectively. As it can noticed computed results reflect the stability conditions given by Eqs. (5.25) and (5.26). It is important to note that when considering fully explicit schemes, the stability is found being prescribed by the mesh size related to the thickness direction, however, when considering the hybrid schema the stability

Table 5.1 – Material properties

E (N/m^2):	$2\ 10^{11}$
ν:	0.25
ρ (kg/m^3):	8000
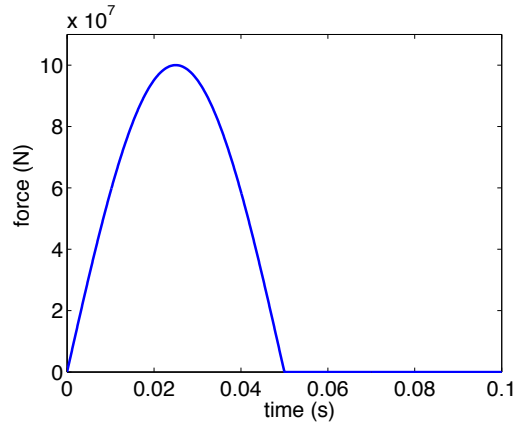
Figure 5.20 – Loading

becomes given by the in-plane characteristic mesh size, that being much larger that the one related to the thickness, integration becomes more more efficient.

To validate the hybrid approach (only in what concerns accuracy and stability, because issues related to computing time savings were addressed in [Bognet *et al.* 2012]), the computed solution is compared with both explicit and implicit 3D finite elements integration with a time step (in the explicit case) guaranteeing the integration stability. The simulation parameters are the ones introduced previously concerning the material properties, and the ones concerning the remaining simulation parameters are indicated in Table 5.2, where N_x, N_y and N_z refer to the number of elements involved in the discretization of directions x, y and z.

Figure 5.23 depicts the time evolution of the vertical displacement w at the central point on segment FG when using different integration schemes. The solution obtained by using the hybrid strategy agrees in minute with the one obtained by using the finite element

Table 5.2 – Simulation parameters

H_x:	$3\,m$
H_y:	$3\,m$
H_z:	$0.1\,m$
N_x:	10
N_y:	2
N_z:	10
Δt:	$10^{-6}\,s$
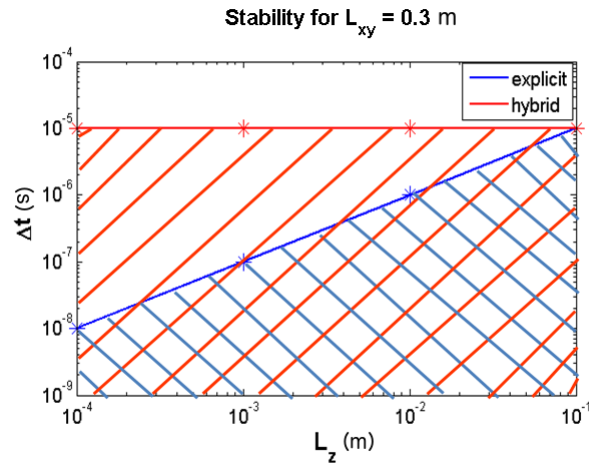
Figure 5.21 – Stability analysis for a given in-plane characteristic mesh size L_{xy}.
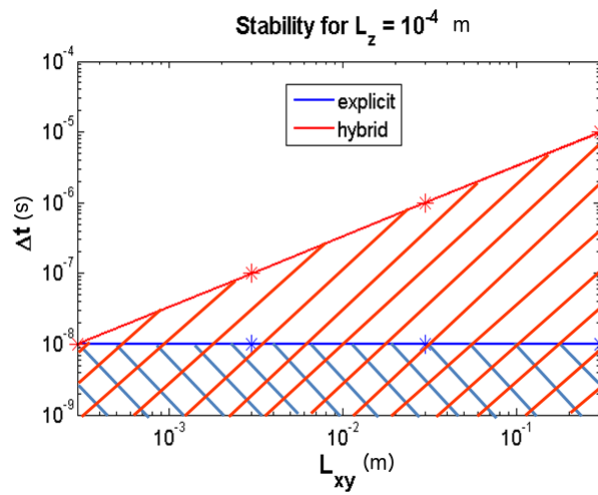
Figure 5.22 – Stability analysis for a given out-of-plane characteristic mesh size L_z.
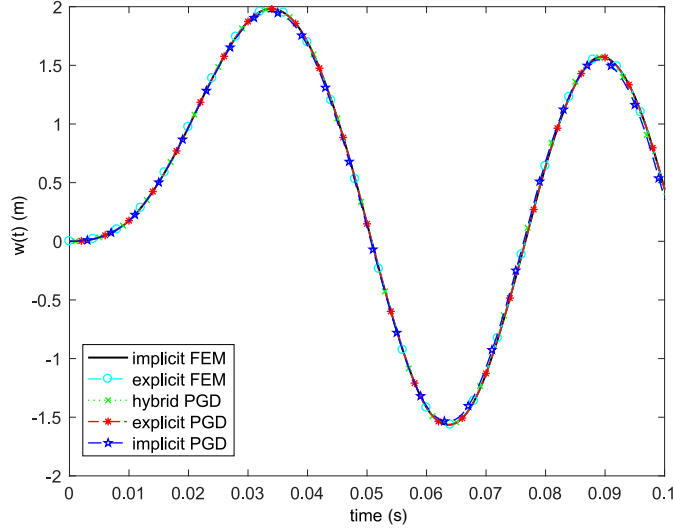
Figure 5.23 – Vertical displacement at the central point of segment FG

method and considered as reference for comparison purposes.

5.2.3.2   Considering richer out-of-plane approximations

In order to check the ability of the proposed technique for addressing richer out-of-plane representations, we consider that the domain depicted in Fig. 5.19 consists now in a laminated composed of 8 anisotropic plies $[0, 45, -45, 90]_S$. The applied force now writes again $\mathbf{F}(\mathbf{t}) = (0, 0, A\sin(\omega t))$ and applies on the face BCGF, with $A = 10^8$ but now with $\omega = 200\pi$, as depicted in Fig. 5.24.

The mechanical properties of the 0 degree-ply are given in Table 5.3, where E is the Young modulus, ν the Poisson coefficient, G the shear modulus and ρ the density. The subscripts indicate respectively the proprieties along the longitudinal direction of the fibers (1), the in-plane transverse direction (2) and the out-of-plane direction (3).

In the present case the elastic constitutive equation becomes orthotropic and using again Voigt notation it reads

$$
\begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ \gamma_{yz} \\ \gamma_{xz} \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{12}}{E_1} & -\frac{\nu_{13}}{E_1} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{23}}{E_2} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{yz} \\ \sigma_{xz} \\ \sigma_{xy} \end{bmatrix}. \tag{5.43}
$$

We compared the solution obtained using the hybrid strategy with the one obtained using implicit finite elements. An explicit finite element solution was not envisaged in this particular case because the too small time step induced by the extremely fine through-the-thickness mesh. The simulation parameters are reported in Tables 5.3 and 5.4.

Figure 5.25 compares the time evolution of the vertical displacement at the middle of segment FG. It can be noticed again that the hybrid strategy provides an excellent solution compared with the implicit finite element considered as reference.
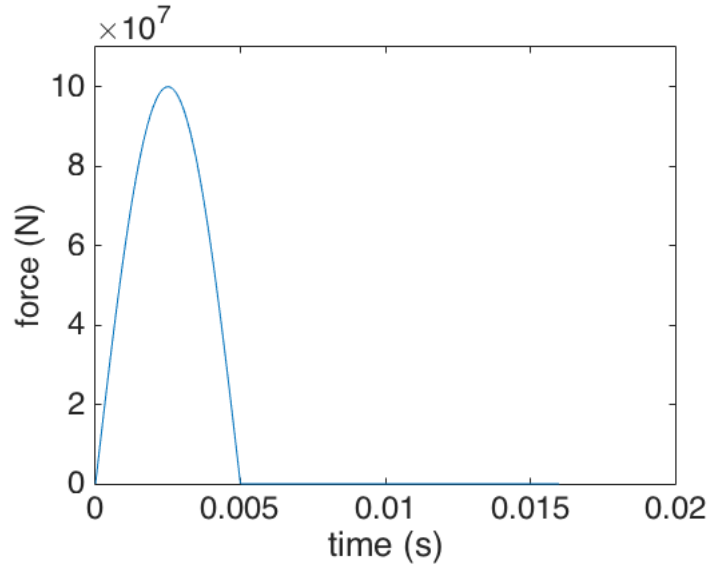
Figure 5.24 – Loading applying in the composite laminate

Table 5.3 – Mechanical properties of the 0 degree-ply

E_1 (N/mm^2):	$120 \cdot 10^3$
E_2 (N/mm^2):	$8.9 \cdot 10^3$
E_3 (N/mm^2):	$8.9 \cdot 10^3$
ν_{12}:	0.35
ν_{13}:	0.35
ν_{23}:	0.32
ν_{21}:	ν_{12}
ν_{31}:	ν_{13}
ν_{32}:	ν_{23}
G_{12} (N/mm^2):	$4.5 \cdot 10^3$
G_{13} (N/mm^2):	$4.5 \cdot 10^3$
G_{23} (N/mm^2):	$5.3 \cdot 10^3$
ρ (kg/m^3):	1750

Table 5.4 – Simulation parameters

H_x:	$250\,mm$
H_y:	$100\,mm$
H_z:	$4\,mm$
N_x:	10
N_y:	2
N_z:	48 (6 elements per ply)
Δt:	$10^{-6}\,s$
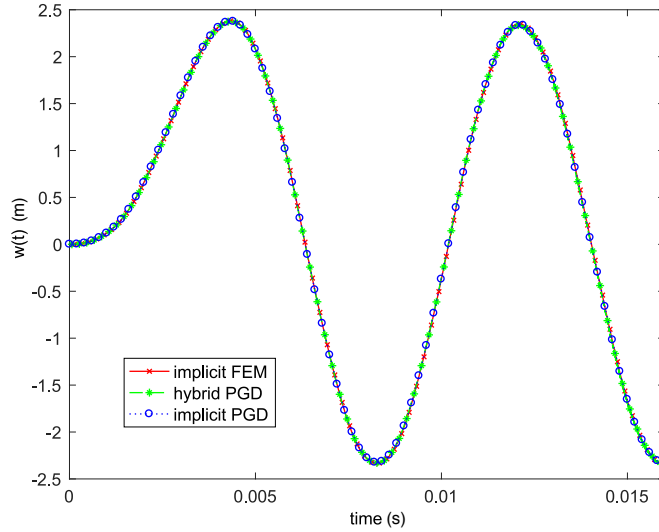
Figure 5.25 – *Vertical displacement at the middle of segment FG*

Finally Fig. 5.26 compares the time evolution of the stress component σ_{zz} at the same location calculated (middle of segment FG) using the finite element method and the implicit/explicit hybrid in-plane-out-of-plane decomposition. Again both solutions match perfectly.

5.2.3.3 Analysis of computational performances

In order to investigate the performances of the proposed technique we perform in this part different analyses. Before, we would like advertising on two facts. First, all computing times are referred to a Matlab implementation on a standard laptop. Thus, computed results allow comparing approaches but not to conclude on absolute performances. Second, for the sake of generality the problem linearity is not taken into account in the sense that at each time step a linear system is solved without taking advantage of the numerous computational profits that linearity offers in the finite element framework. This conservative approach allows extending the main conclusions to the nonlinear case.

We consider again the problem defined in the domain Ω depicted in Fig. 5.19, with $H_x = H_y = 3\,m$, $H_z = 0.1\,m$ and with the material properties defined in Table 5.1, considering the same boundary conditions than in Section 5.2.3.1 and the same loading, the last illustrated in Fig. 5.20. In the analyses here addressed, the PGD constructor stoping criterion is set to $\epsilon = \tilde{\epsilon} = 10^{-6}$.

First, we compare the hybrid PGD method with its fully implicit counterpart. The three different meshes defined in Table 5.5 are considered, where again N_x, N_y and N_z refer to the number of elements involved in the discretization of directions x, y and z respectively. For each mesh we compare the computing time employed by both the hybrid and the fully implicit PGD discretizations to solve the problem in the time interval $[0, 400\Delta t]$, with the time-step $\Delta t = \frac{10^{-5}}{3}\,s$ for all the simulations.

Results presented in Fig. 5.27 prove that, as expected, when using the same time-step the hybrid method proceeds faster than the implicit one. Later, in order to take advantage of the superior stability of fully implicit discretizations, time-steps will be selected differently for ensuring an equivalent accuracy, in order to compare computing costs in a
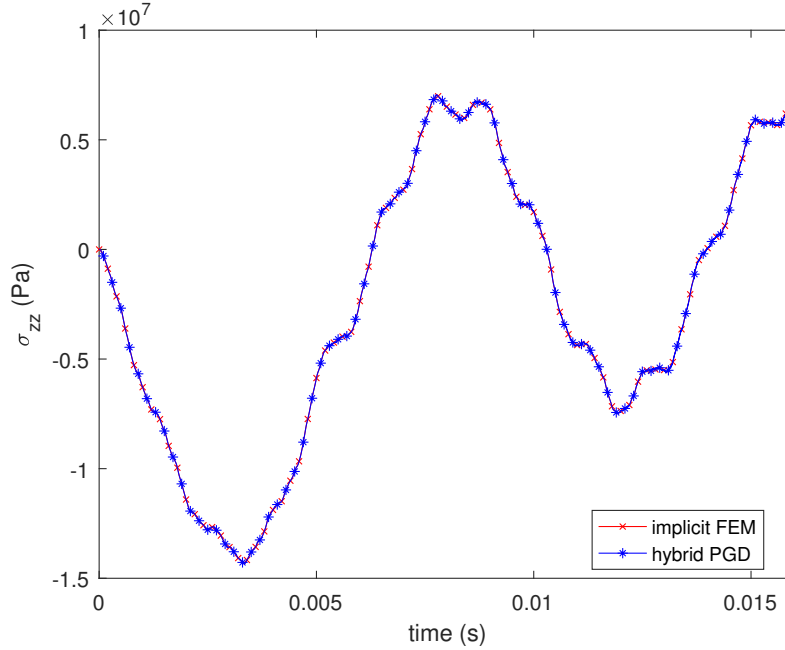
Figure 5.26 – Stress component $\sigma_{zz}(t)$ at the central point of segment FG. Implicit and hybrid-based solutions are almost superimposed.

Table 5.5 – Meshes considered in the analysis of computational performances depicted in Fig. 5.27.

	Mesh 1	Mesh 2	Mesh 3
N_x:	10	20	30
N_y:	10	20	30
N_z:	100	100	100

more appropriate manner.

Now, we perform a comparison between the three PGD formulations (explicit, hybrid and implicit) in the time interval $[0, 400\Delta t]$, whith $\Delta t = 10^{-7}\,s$ to ensure the stability of the explicit time integration. Results for the three meshes in Table 5.6 are presented in Figure 5.28. As expected the computational cost of the hybrid formulation is in between the one of explicit and implicit time integrations.

We have already proved in Figs. 5.21 and 5.22 that the stability domain of the hybrid formulation does not depend on the mesh size associated with the thickness direction, so that the hybrid simulation proceeds faster than the one performed using a fully explicit formulation by using a larger time-step in the hybrid integration.

In order to prove it, we perform a simulation using 'Mesh 2" in Table 5.5, in the time interval $[0, 4]\,ms$. We use as time steps for the explicit and the hybrid methods respectively $\Delta t_{ex} = 10^{-7}\,s$ and $\Delta t_{hy} = 10^{-5}\,s$, ensuring the stability of both schemes. Results shown in Fig. 5.29 reveal as expected that the higher time-step considered in the hybrid integration induces significant computing time savings.

196
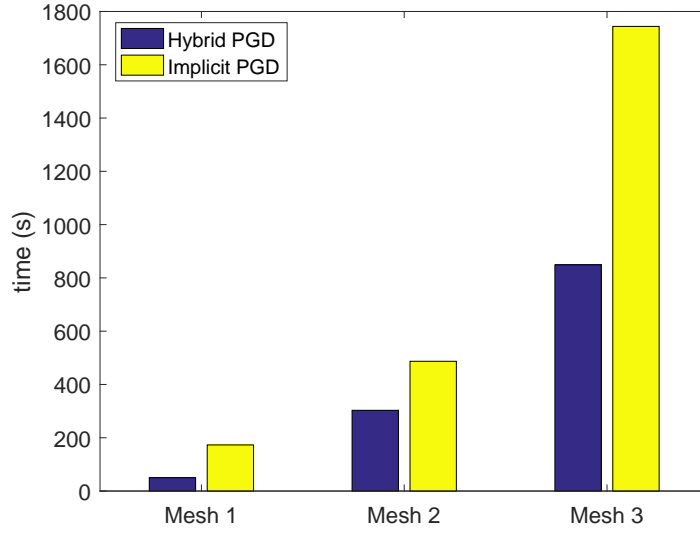
Figure 5.27 – Hybrid versus implicit PGD formulations.

Table 5.6 – Meshes considered in the analysis of computational performances depicted in 5.28.

	Mesh 4	Mesh 5	Mesh 6
N_x:	10	20	30
N_y:	10	20	30
N_z:	50	50	50
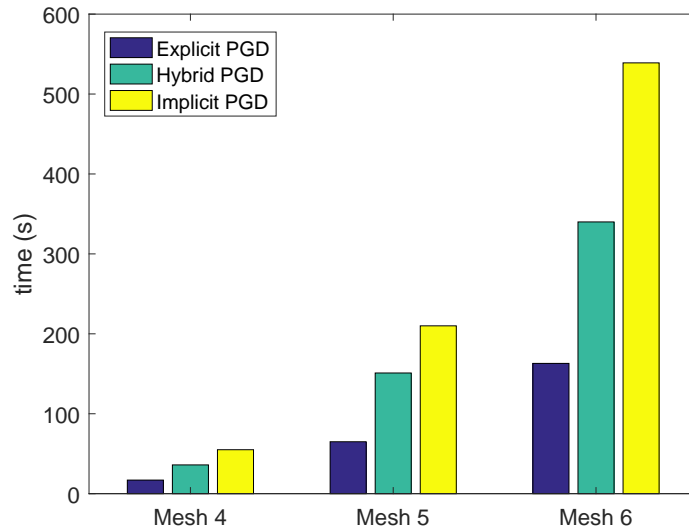
Figure 5.28 – Comparing explicit, hybrid and implicit PGD formulations.
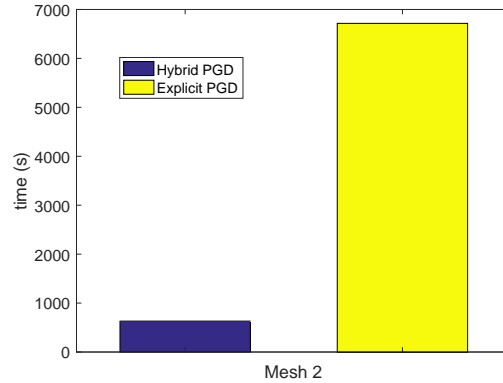
Figure 5.29 – Hybrid versus explicit PGD formulations.

Table 5.7 – Mesh used in the results described in Figure 5.30 and Table 5.8.

	Mesh 7
N_x:	10
N_y:	10
N_z:	90

The last analysis aims at taking advantage of the superior stability performances of the implicit formulation, that a priori can use larger time-steps that the ones of explicit and hybrid formulations that are only conditionally stables. However, here not only stability issues are addressed but also the accuracy of the computed solutions.

Thus, in the present numerical analysis we consider the mesh defined Table 5.7 and the time interval $[0, T]$ with $T = 6\,ms$, and consider as *reference solution* the one computed using an explicit FEM scheme with a very fine time-step $\Delta t = 10^{-7}\,s$, ensuring both stability and accuracy, both performances having been checked. Figure 5.30 compares the computational cost related to FEM and PGD explicit time integrations. As expected the separated representation involved in the PGD formulation allows better performances.

Then the problem is solved using first the hybrid scheme with a time-step $\Delta t = 2 \cdot 10^{-5}\,s$ that ensures its stability and implicit (PGD and FEM) time integrations using higher time-steps. For each solution we consider the computational cost as well as the error E with respect to the reference solution, computed from the relative $L2$ norm:

$$E = \frac{\left(\int_0^T \int_\Omega \left(\mathbf{u} - \mathbf{u}_{ref}\right)^2 d\mathbf{x} dt\right)^{\frac{1}{2}}}{\left(\int_0^T \int_\Omega \mathbf{u}_{ref}^2 d\mathbf{x} dt\right)^{\frac{1}{2}}}. \tag{5.44}$$

Table 5.8 compares the different solutions, proving that: (i) implicit PGD and FEM integrations lead to almost the same solutions, being the ones related to PGD less computationally expensive; and (ii) implicit simulations related to the same computational cost than hybrid simulation produce larger errors, for the analyzed case.
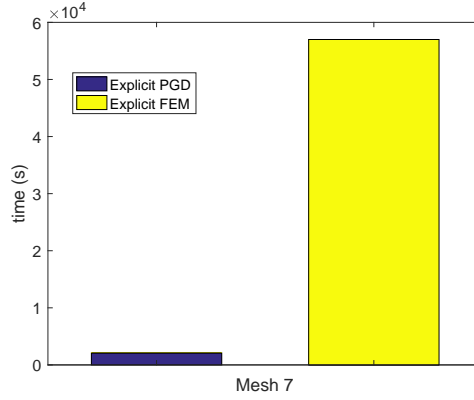
Figure 5.30 – Explicit PGD versus explicit FEM.

Table 5.8 – Hybrid and implicit PGD integrations versus a standard implicit finite element formulation (using different time-steps).

		Time	Error
$\Delta t = 2 \cdot 10^{-5} \, s$	Hybrid PGD:	$65 \, s$	$8.3 \cdot 10^{-3}$
	Implicit PGD:	$203 \, s$	$8.4 \cdot 10^{-3}$
	Implicit FEM:	$746 \, s$	$8.4 \cdot 10^{-3}$
$\Delta t = 4 \cdot 10^{-5} \, s$	Implicit PGD:	$185 \, s$	$1.66 \cdot 10^{-2}$
	Implicit FEM:	$363 \, s$	$1.66 \cdot 10^{-2}$
$\Delta t = 6 \cdot 10^{-5} \, s$	Implicit PGD:	$145 \, s$	$2.47 \cdot 10^{-2}$
	Implicit FEM:	$236 \, s$	$2.47 \cdot 10^{-2}$
$\Delta t = 8 \cdot 10^{-5} \, s$	Implicit PGD:	$88 \, s$	$3.27 \cdot 10^{-2}$
	Implicit FEM:	$174 \, s$	$3.27 \cdot 10^{-2}$
$\Delta t = 10^{-4} \, s$	Implicit PGD:	$63 \, s$	$4.05 \cdot 10^{-2}$
	Implicit FEM:	$146 \, s$	$4.05 \cdot 10^{-2}$

5.2.4 Conclusion

A hybrid time discretization scheme for solving 3D dynamical problems defined in degenerated domains is proposed, that is, domains in which one of its characteristic dimensions is much smaller that the other ones, as it is the case when considering plates or shells.

A first complexity reduction is attained by considering the in-plane-out-of-plane separated representation that allows reducing the original 3D complexity to the one characteristic of 2D plate or shells models, even if the computed solution is fully 3D and any hypothesis is introduced. Such separated representation allows the use of extremely fine descriptions along the thickness direction.

However, such decomposition when combined with explicit time integrations has a major handicap, the too small element length involved in the thickness direction implies an extremely small time step for ensuring stability. Such a drawback is circumvented by using an implicit (unconditionally stable) through-the-thickness operators whereas a standard explicit scheme is considered for treating the in-plane operators. By doing that, the stability is dictated by the in-plane mesh size, much coarser than the one employed in the thickness direction, alleviating the stability constraints. It is important to note, that even if a part of the whole scheme remains implicit, it only affects one dimension and then its solution is extremely fast and cheap.

The inclusion of progressive damage models combined with dynamical effects constitutes a work in progress, where the separated representations seems an appealing option to better represent damage effects along the laminate thickness, and where explicit time integrations are usually employed in industrial applications.

5.3 Simulating Squeeze Flows in Multiaxial Laminates: Towards Fully 3D Mixed Formulations

In what follows we first address the penalty and mixed formulations of the Stokes flow in a narrow gap, that can be easily generalized to stratified flows. Then the flow of multi axial laminates making use of the Ericksen fluid flow model at the ply level is considered. In this last case the penalty formulation related to both the fiber inextensibility and the flow incompressibility is substituted in favor of a mixed formulation making use of two Lagrange multipliers, the first related to the inextensibility constraint and the second one to the flow incompressibility. Such a richer description is needed to evaluate the fiber tension, crucial to predict defects related to its compression. On the other hand the rheological characterization of multiaxial laminates is performed by calculating the compression force to be applied for obtaining a given squeeze rate. For that purpose, it is important calculating the stress tensor in the fluid, and when using a penalty formulation the calculation of the pressure field remains a tricky issue. These facts justify the use of a mixed formulation instead of the penalized one previously considered in our former works, formulation that was retained in [Ghnatios *et al.* 2016] for circumventing the issues related to the LBB stability condition.

5.3.1 3D modeling of Stokes flow in narrow gaps

5.3.1.1 In-plane-out-of-plane separated representation

The in-plane-out-of-plane separated representation allows the solution of full 3D models defined in plate geometries with a computational complexity characteristic of 2D simula-

tions. This separated representation allows independent representations of the in-plane and
the thickness fields dependencies. The main idea lies in the separated representation of the
velocity field by using functions depending on the in-plane coordinates $\mathbf{x} = (x, y)$, $\mathbf{P}_i^j(\mathbf{x})$,
and others depending on the thickness direction z, $\mathbf{T}_i^j(z)$, according to:

$$
\begin{pmatrix} \mathbf{v}(\mathbf{x}, z) \\ p(\mathbf{x}, z) \\ \tau(\mathbf{x}, z) \end{pmatrix} = \begin{pmatrix} u(\mathbf{x}, z) \\ v(\mathbf{x}, z) \\ w(\mathbf{x}, z) \\ p(\mathbf{x}, z) \\ \tau(\mathbf{x}, z) \end{pmatrix} \approx \begin{pmatrix} \sum\limits_{i=1}^{N} P_i^1(\mathbf{x}) \cdot T_i^1(z) \\ \sum\limits_{i=1}^{N} P_i^2(\mathbf{x}) \cdot T_i^2(z) \\ \sum\limits_{i=1}^{N} P_i^3(\mathbf{x}) \cdot T_i^3(z) \\ \sum\limits_{i=1}^{N} P_i^p(\mathbf{x}) \cdot T_i^p(z) \\ \sum\limits_{i=1}^{N} P_i^\tau(\mathbf{x}) \cdot T_i^\tau(z) \end{pmatrix} = \sum_{i=1}^{N} \mathbf{P}_i(\mathbf{x}) \circ \mathbf{T}_i(z), \quad (5.45)
$$

which leads to a separated representation of the strain rate. When introduced into the flow
problem weak form it allows the calculation of functions $P_i(\mathbf{x})$ by solving the corresponding
2D equations and functions $T_i(z)$ by solving the associated 1D equations, as described later.

Eq. (5.45) can be rewritten in the compact form

$$
\mathbf{v}(\mathbf{x}, z) \approx \sum_{i=1}^{N} \mathbf{P}_i^v(\mathbf{x}) \circ \mathbf{T}_i^v(z) = \begin{pmatrix} \sum\limits_{i=1}^{N} P_i^1(\mathbf{x}) \cdot T_i^1(z) \\ \sum\limits_{i=1}^{N} P_i^2(\mathbf{x}) \cdot T_i^2(z) \\ \sum\limits_{i=1}^{N} P_i^3(\mathbf{x}) \cdot T_i^3(z) \end{pmatrix}, \quad (5.46)
$$

where "∘" denotes the entry-wise or Hadamard's product.

Remark 1. If $\mathbf{a}$ and $\mathbf{b}$ are vectors of the same dimension, vector $\mathbf{c}$, defined from $\mathbf{c} = \mathbf{a} \circ \mathbf{b}$,
has as components $c_i = a_i \cdot b_i$. If $\mathbf{a}$ and $\mathbf{b}$ are second order tensors with the same size,
tensor $\mathbf{c}$, defined from $\mathbf{c} = \mathbf{a} \circ \mathbf{b}$, has components $c_{ij} = a_{ij} \cdot b_{ij}$ (no sum with respect to the
repeated indexes). In this case it results $\mathbf{a} : \mathbf{b} = c$, with the scalar c given by $c = a_{ij} \cdot b_{ij}$
considering sum with respect to the repeated indexes (Einstein's summation convention).

Using this notation in (5.45), the velocity gradient $\nabla \mathbf{v}(\mathbf{x}, z)$ can be written as:

$$
\nabla \mathbf{v} = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix} \approx \sum_{i=1}^{N} \begin{pmatrix} \frac{\partial P_i^1}{\partial x} & \frac{\partial P_i^1}{\partial y} & P_i^1 \\ \frac{\partial P_i^2}{\partial x} & \frac{\partial P_i^2}{\partial y} & P_i^2 \\ \frac{\partial P_i^3}{\partial x} & \frac{\partial P_i^3}{\partial y} & P_i^3 \end{pmatrix} \circ \begin{pmatrix} T_i^1 & T_i^1 & \frac{\partial T_i^1}{\partial z} \\ T_i^2 & T_i^2 & \frac{\partial T_i^2}{\partial z} \\ T_i^3 & T_i^3 & \frac{\partial T_i^3}{\partial z} \end{pmatrix} =
$$

$$
\sum_{i=1}^{N} \mathbb{P}_i(\mathbf{x}) \circ \mathbb{T}_i(z). \quad (5.47)
$$

The solution of full 3D Stokes problem within the in-plane-out-of-plane separated rep-
resentation is revisited in the next sections.

5.3.1.2 Flow model

The Stokes flow model is defined in $\Xi = \Omega \times \mathcal{I}$, $\Omega \subset \mathbb{R}^2$ and $\mathcal{I} \subset \mathbb{R}$, and for an incompressible fluid, in absence of inertia and mass terms reduces to:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \\ \boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D} \\ \nabla \cdot \mathbf{v} = 0 \end{cases}, \tag{5.48}$$

where $\boldsymbol{\sigma}$ is the Cauchy's stress tensor, $\mathbf{I}$ the unit tensor, η the fluid viscosity, p the pressure (Lagrange multiplier associated with the incompressibility constraint) and the rate of strain tensor $\mathbf{D}$ defined as

$$\mathbf{D} = \frac{\nabla \mathbf{v} + (\nabla \mathbf{v})^T}{2}. \tag{5.49}$$

The pressure in-plane-out-of-plane separated representation writes

$$p = \sum_{i=1}^{N} P_i^p(\mathbf{x}) \cdot T_i^p(z). \tag{5.50}$$

In what follows for the sake of simplicity the dependency of in-plane functions on $\mathbf{x}$ an the one of out-of-plane functions on z will be omitted.

The weak form of the coupled velocity-pressure Stokes problem, for both a test velocity $\mathbf{v}^*$ and a test pressure $\mathbf{p}^*$, the first vanishing on the boundary in which the velocity is prescribed, and assuming null tractions in the remaining part of the domain boundary, can be written as

$$\int_{\Omega \times \mathcal{I}} \left(-p\mathrm{Tr}(\mathbf{D}^*) + 2\eta\mathbf{D}^* : \mathbf{D} \right) \, d\mathbf{x} \, dz = 0, \tag{5.51}$$

$$\int_{\Omega \times \mathcal{I}} -p^*\mathrm{Tr}(\mathbf{D}) \, d\mathbf{x} \, dz = 0, \tag{5.52}$$

where Eqs. (5.51) and (5.52) make reference to the linear momentum and mass balances respectively.

Following the developments reported in the Appendix B, previous balances can be rewritten as

$$2\eta\mathbf{D}^* : \mathbf{D} \approx \frac{\eta}{2} \sum_{j=1}^{N} \sum_{k=1}^{4} \left(\mathbb{A}_{jk}^*(\mathbf{x}) : \mathbb{B}_{jk}(z) + \mathbb{A}_{jk}(\mathbf{x}) : \mathbb{B}_{jk}^*(z) \right), \tag{5.53}$$

$$p\mathrm{Tr}(\mathbf{D}^*) \approx \sum_{i=1}^{N} P_i^p \cdot T_i^p \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^*, \tag{5.54}$$

and

$$p^*\mathrm{Tr}(\mathbf{D}) \approx \sum_{i=1}^{N} (P^p \cdot T^p)^* \left(\frac{\partial P_i^1}{\partial x} T_i^1 + \frac{\partial P_i^2}{\partial y} T_i^2 + P_i^3 \frac{\partial T_i^3}{\partial z} \right). \tag{5.55}$$

5.3.1.3 Separated representation constructor

The construction of the solution separated representation is performed incrementally, a term of the sum at each iteration. Thus, supposing that at iteration $n-1$, $n \geq 1$, the first $n-1$ terms of both velocity and pressure separated representations were already computed

$$\mathbf{v}^{n-1}(\mathbf{x}, z) = \sum_{i=1}^{n-1} \mathbf{P}_i^v(\mathbf{x}) \circ \mathbf{T}_i^v(z), \tag{5.56}$$

$$p^{n-1}(\mathbf{x}, z) = \sum_{i=1}^{n-1} P_i^p(\mathbf{x}) \cdot T_i^p(z), \tag{5.57}$$

the terms involved in the weak form (5.51) and (5.52) are:

$$\mathbf{D}^* : \mathbf{D}^{n-1} = \frac{1}{4} \sum_{j=1}^{n-1} \sum_{k=1}^{4} \left(\mathbb{A}_{jk}^*(\mathbf{x}) : \mathbb{B}_{jk}(z) + \mathbb{A}_{jk}(\mathbf{x}) : \mathbb{B}_{jk}^*(z) \right), \tag{5.58}$$

$$p^{n-1} \mathrm{Tr}(\mathbf{D}^*) \approx \sum_{i=1}^{n-1} P_i^p \cdot T_i^p \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^*, \tag{5.59}$$

and

$$p^* \mathrm{Tr}(\mathbf{D}^{n-1}) \approx \sum_{i=1}^{n-1} (P^p \cdot T^p)^* \left(\frac{\partial P_i^1}{\partial x} T_i^1 + \frac{\partial P_i^2}{\partial y} T_i^2 + P_i^3 \frac{\partial T_i^3}{\partial z} \right), \tag{5.60}$$

respectively. The indexes affecting test functions will be detailed below.

When looking for the improved velocity field $\mathbf{v}^n(\mathbf{x}, z)$ at iteration n

$$\mathbf{v}^n(\mathbf{x}, z) = \sum_{i=1}^{n} \mathbf{P}_i^v(\mathbf{x}) \circ \mathbf{T}_i^v(z) = \mathbf{v}^{n-1}(\mathbf{x}, z) + \mathbf{P}_n^v(\mathbf{x}) \circ \mathbf{T}_n^v(z), \tag{5.61}$$

we consider the test function $\mathbf{v}^*(\mathbf{x}, z)$

$$\mathbf{v}^* = \mathbf{P}^{v*} \circ \mathbf{T}_n^v + \mathbf{P}_n^v \circ \mathbf{T}^{v*}, \tag{5.62}$$

that implies

$$\nabla \mathbf{v}^* = \mathbb{P}^* \circ \mathbb{T}_n + \mathbb{P}_n \circ \mathbb{T}^*. \tag{5.63}$$

When looking for the improved pressure field $p^n(\mathbf{x}, z)$ at iteration n

$$p^n(\mathbf{x}, z) = \sum_{i=1}^{n} P_i^p(\mathbf{x}) \cdot T_i^p(z) = p^{n-1}(\mathbf{x}, z) + P_n^p(\mathbf{x}) \cdot T_n^p(z), \tag{5.64}$$

we consider the test function

$$p^* = P^{p*} \cdot T_n^p + P_n^p \cdot T^{p*}. \tag{5.65}$$

In that case it results

$$4\mathbf{D}^* : \mathbf{D}^n = \sum_{j=1}^{n-1} \sum_{k=1}^{4} \left(\mathbb{A}_{jk}^*(\mathbf{x}) : \mathbb{B}_{jk}(z) + \mathbb{A}_{jk}(\mathbf{x}) : \mathbb{B}_{jk}^*(z) \right) +$$

$$\sum_{k=1}^{4} \left(\mathbb{A}_{nk}^*(\mathbf{x}) : \mathbb{B}_{nk}(z) + \mathbb{A}_{nk}(\mathbf{x}) : \mathbb{B}_{nk}^*(z) \right), \tag{5.66}$$

$$p^n \mathrm{Tr}(\mathbf{D}^*) \approx \sum_{i=1}^{n-1} P_i^p \cdot T_i^p \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^* +$$

$$P_n^p \cdot T_n^p \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^*, \tag{5.67}$$

and

$$p^* \mathrm{Tr}(\mathbf{D}^n) \approx \sum_{i=1}^{n-1} (P^p \cdot T^p)^* \left(\frac{\partial P_i^1}{\partial x} T_i^1 + \frac{\partial P_i^2}{\partial y} T_i^2 + P_i^3 \frac{\partial T_i^3}{\partial z} \right) +$$

$$(P^p \cdot T^p)^* \left(\frac{\partial P_n^1}{\partial x} T_n^1 + \frac{\partial P_n^2}{\partial y} T_n^2 + P_n^3 \frac{\partial T_n^3}{\partial z} \right), \tag{5.68}$$

where

$$\left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^* = \left(\frac{\partial P^{1*}}{\partial x} T_n^1 + \frac{\partial P^{2*}}{\partial y} T_n^2 + P^{3*} \frac{\partial T_n^3}{\partial z} \right) +$$

$$\left(\frac{\partial P_n^1}{\partial x} T^{1*} + \frac{\partial P_n^2}{\partial y} T^{2*} + P_n^3 \frac{\partial T^{3*}}{\partial z} \right), \tag{5.69}$$

and

$$(P^p \cdot T^p)^* = P^{p*} \cdot T_n^p + P_n^p \cdot T^{p*}. \tag{5.70}$$

Thus the problem weak form (5.51) and (5.52) writes at iteration n:

$$\int\limits_{\Omega \times \mathcal{I}} -P_n^p \cdot T_n^p \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^* \, d\mathbf{x} \, dz +$$

$$\int\limits_{\Omega \times \mathcal{I}} \frac{\eta}{2} \left(\sum_{k=1}^{4} \mathbb{A}_{nk}^*(\mathbf{x}) : \mathbb{B}_{nk}(z) + \mathbb{A}_{nk}(\mathbf{x}) : \mathbb{B}_{nk}^*(z) \right) \, d\mathbf{x} \, dz =$$

$$\int\limits_{\Omega \times \mathcal{I}} \sum_{j=1}^{n-1} P_i^4 \cdot T_i^4 \left(\frac{\partial P^1}{\partial x} T^1 + \frac{\partial P^2}{\partial y} T^2 + P^3 \frac{\partial T^3}{\partial z} \right)^* \, d\mathbf{x} \, dz -$$

$$\int\limits_{\Omega \times \mathcal{I}} \frac{\eta}{2} \left(\sum_{j=1}^{n-1} \sum_{k=1}^{4} \left(\mathbb{A}_{jk}^*(\mathbf{x}) : \mathbb{B}_{jk}(z) + \mathbb{A}_{jk}(\mathbf{x}) : \mathbb{B}_{jk}^*(z) \right) \right) \, d\mathbf{x} \, dz, \tag{5.71}$$

and

$$\int\limits_{\Omega \times \mathcal{I}} -(P^p \cdot T^p)^* \left(\frac{\partial P_n^1}{\partial x} T_n^1 + \frac{\partial P_n^2}{\partial y} T_n^2 + P_n^3 \frac{\partial T_n^3}{\partial z} \right) \, d\mathbf{x} \, dz =$$

$$\int\limits_{\Omega \times \mathcal{I}} \sum_{i=1}^{n-1} (P^p \cdot T^p)^* \left(\frac{\partial P_i^1}{\partial x} T_i^1 + \frac{\partial P_i^2}{\partial y} T_i^2 + P_i^3 \frac{\partial T_i^3}{\partial z} \right) \, d\mathbf{x} \, dz. \tag{5.72}$$

The extended weak forms (5.71) and (5.72) become nonlinear because it involves the product of unknown functions $\mathbf{P}_n$ and $\mathbf{T}_n$. Thus a linearization strategy becomes necessary, the simplest one being an alternating fixed point algorithm that proceeds as follows:

1. Assuming functions $\mathbf{P}_n(\mathbf{x})$ are known (arbitrarily chosen at the first iteration of the nonlinear iteration) matrices $\mathbb{A}_{jk}^*$ and $\mathbb{F}_j^*$, as well as functions $\left(\frac{\partial P^{1*}}{\partial x} T_n^1 + \frac{\partial P^{2*}}{\partial y} T_n^2 + P^{3*} \frac{\partial T_n^3}{\partial z} \right)$ and P^{p*} vanish. Being all functions depending on $\mathbf{x}$ known, integrals in Ω in (5.71) and (5.72) can be calculated. Thus, it finally results in a one dimensional linear problem that involves the four scalar functions involved in $\mathbf{T}_n(z)$, $T_n^1(z)$, $T_n^2(z)$, $T_n^3(z)$ and $T_n^p(z)$.

2. Then, with the just computed function $\mathbf{T}_n(z)$, and with $\mathbb{B}_{jk}^*$, $\mathbb{G}_j^*$, T^{p*} and $\left(\frac{\partial P_n^1}{\partial x} T^{1*} + \frac{\partial P_n^2}{\partial y} T^{2*} + P_n^3 \frac{\partial T^{3*}}{\partial z} \right)$ vanishing, one can proceed to integrate Eqs. (5.71) and (5.72) in $\mathcal{I}$. It finally results in a two-dimensional linear problem for the unknown function $\mathbf{P}_n(\mathbf{x})$ that involves the four scalar functions $P_n^1(\mathbf{x})$, $P_n^2(\mathbf{x})$, $P_n^3(\mathbf{x})$ and $P_n^p(\mathbf{x})$.

3. The convergence is checked by comparing functions $\mathbf{P}_n$ and $\mathbf{T}_n$ between two consecutive iterations of the nonlinear solver. If both functions are small enough they are used to update both velocity and pressure fields

$$\begin{pmatrix} \mathbf{v}(\mathbf{x}, z) \\ p(\mathbf{x}, z) \end{pmatrix} = \sum_{i=1}^{n} \mathbf{P}_i(\mathbf{x}) \circ \mathbf{T}_i(z). \tag{5.73}$$

If the convergence is not attained, one returns to step 1 with the calculated functions $\mathbf{P}_n$ to re-compute $\mathbf{T}_n$

Because of the one-dimensional large scale variation present in the laminate thickness direction one can employ extremely detailed descriptions along the thickness direction without sacrificing the computational efficiency of the 3D solution procedure.

5.3.1.4 Flow in a laminate

Consider a laminate composed of $\mathcal{P}$ layers in which each layer involves a linear and isotropic viscous fluid of viscosity η_i, thus the extended Stokes flow problem in its weak form involves the dependence of the viscosity along the thickness direction.

If H is the total laminate thickness, and assuming for the sake of simplicity and without loss of generality that all the plies have the same thickness h, it results $h = \frac{H}{\mathcal{P}}$. Now, from the characteristic function of each ply $\chi_i(z)$, $i = 1, \cdots, \mathcal{P}$:

$$\chi_i(z) = \begin{cases} 1 & \text{if } (i-1)h \le z < ih \\ 0 & \text{elsewehere} \end{cases}, \tag{5.74}$$

the viscosity reads

$$\eta(\mathbf{x}, z) = \sum_{i=1}^{\mathcal{P}} \eta_i \cdot \chi_i(z), \tag{5.75}$$

where it is assumed, again without loss of generality, that the viscosity does not evolve in the plane, i.e. $\eta_i(\mathbf{x}) = \eta_i$.

This decomposition is fully compatible with the velocity-pressure separated representation (5.45) and with the in-plane-out-of-plane decomposition considered for solving Eqs. (5.71) and (5.72).

5.3.2 Ericksen fluid flow model in a laminate

The case of a prepreg ply reinforced by continuous fibres oriented along direction $\mathbf{p}^T = (p_x, p_y, 0)$, $\|\mathbf{p}\| = 1$, is analyzed here. It is assumed that the thermoplastic resin exhibits Newtonian behaviour. Thus the velocity $\mathbf{v}(\mathbf{x}, z)$ of the equivalent anisotropic fluid must satisfy the incompressibility and inextensibility constraints

$$\nabla \cdot \mathbf{v} = 0, \tag{5.76}$$

and

$$\mathbf{p}^T \cdot \nabla \mathbf{v} \cdot \mathbf{p} = 0, \tag{5.77}$$

respectively. Expression (5.77) can be rewritten using tensor notation as $\nabla \mathbf{v} : \mathbf{a} = 0$, where the second order orientation tensor $\mathbf{a}$ is defined from $\mathbf{a} = \mathbf{p} \cdot \mathbf{p}^T = \mathbf{p} \otimes \mathbf{p}$.

The orientation tensor $\mathbf{a}$ has only planar components (the out-of-plane fiber orientation can be neglected in the case of laminates), it is symmetric and of unit trace, i.e.

$$\mathbf{a} = \begin{pmatrix} a_{xx} & a_{xy} & 0 \\ a_{yx} & a_{yy} & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \mathcal{A} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{pmatrix}, \tag{5.78}$$

where $\mathcal{A}$ represents the plane component of the orientation tensor $\mathbf{a}$, $a_{xy} = a_{yx}$ (i.e. $\mathcal{A} = \mathcal{A}^T$) and $a_{yy} = 1 - a_{xx}$.

The simplest expression of the Ericksen's constitutive equation [Ericksen 1959] can be written in the compact form as follows

$$\boldsymbol{\sigma} = -p\mathbf{I} + \tau\mathbf{a} + 2\eta_T\mathbf{D} + 2(\eta_L - \eta_T)(\mathbf{D} \cdot \mathbf{a} + \mathbf{a} \cdot \mathbf{D}), \tag{5.79}$$

that is then introduced into the linear momentum balance

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0}. \tag{5.80}$$

In Eq. (5.79) p and τ represents respectively the Lagrange multipliers related to the incompressibility and inextensibility constraints, and η_L and η_T the longitudinal and transverse shear viscosities respectively.

By separating both the pressure and the fiber tension fields using an in-plane-out-of-plane-separated representation

$$p(\mathbf{x}, z) = \sum_{i=1}^{N} P_i^p(\mathbf{x}) \cdot T_i^p(z), \tag{5.81}$$

and

$$\tau(\mathbf{x}, z) = \sum_{i=1}^{N} P_i^\tau(\mathbf{x}) \cdot T_i^\tau(z), \tag{5.82}$$

it allows accurate calculations of both the fiber tension and the pressure field. This information can be used to predict either fibers buckling or forces acting on the squeezed boundary.

The weak form for a test velocity $\mathbf{v}^*(\mathbf{x}, z)$ vanishing at the boundary in which velocity is prescribed, a test pressure $p^*(\mathbf{x}, z)$ and a test fiber tension $\tau^*(\mathbf{x}, z)$, assuming null tractions in the remaining part of the domain boundary can be expressed as

$$\int_{\Omega \times \mathcal{I}} \mathbf{D}^* : \boldsymbol{\sigma} \, d\mathbf{x} \, dz = 0, \tag{5.83}$$

$$\int_{\Omega \times \mathcal{I}} p^* \mathbf{D} : \mathbf{I} \, d\mathbf{x} \, dz = 0, \tag{5.84}$$

and

$$\int_{\Omega \times \mathcal{I}} \tau^* \mathbf{D} : \mathbf{a} \, d\mathbf{x} \, dz = 0. \tag{5.85}$$

By introducing the Ericksen constitutive equation (5.79), Eq. (5.83) can be written as

$$\int_{\Omega \times \mathcal{I}} \mathbf{D}^* : \boldsymbol{\sigma} \, d\mathbf{x} \, dz =$$

$$\int\limits_{\Omega \times \mathcal{I}} \mathbf{D}^* : \left(-p\mathbf{I} + \tau \mathbf{a} + \eta_T \mathbf{D} + \tilde{\eta} \left(\mathbf{D} \cdot \mathbf{a} + \mathbf{a} \cdot \mathbf{D}\right)\right) \ d\mathbf{x} \ dz = 0, \tag{5.86}$$

with $\tilde{\eta} = \eta_L - \eta_T$.

At this stage the in-plane-out-of-plane separated representation constructor of $\mathbf{v}(\mathbf{x}, z)$, $p(\mathbf{x}, z)$ and $\tau(\mathbf{x}, z)$ proceeds as described in the previous section.

Remark 2. If $\mathbf{a} = \mathbf{0}$ this formulation reduced to one related to the Stokes flow problem.

Remark 3. Laminates can be addressed by associating with each ply the planar fiber orientation $\mathbf{p}_i(\mathbf{x})$, with its out-of-plane component vanishing, from which the associated orientation tensor $\mathbf{a}_i(\mathbf{x})$ results in $\mathbf{a}_i(\mathbf{x}) = \mathbf{p}_i(\mathbf{x}) \otimes \mathbf{p}_i(\mathbf{x})$. Using again the characteristic function of the i-ply, $\chi_i(z)$, $i = 1, \cdots, \mathcal{P}$, the orientation tensor in the laminate, $\mathbf{a}(\mathbf{x}, z)$, can be expressed as

$$\mathbf{a}(\mathbf{x}, z) = \sum_{i=1}^{\mathcal{P}} \mathbf{a}_i(\mathbf{x}) \chi_i(z). \tag{5.87}$$

Remark 4. If the fiber orientation is constant in each plane, then the laminate orientation tensor can be expressed as

$$\mathbf{a}(z) = \sum_{i=1}^{\mathcal{P}} \mathbf{a}_i \chi_i(z). \tag{5.88}$$

5.3.3 Revisiting fully penalized formulations.

Both Stokes and Ericksen flows have been successfully implemented by means of penalized formulations involving both, the pressure, p, and the fiber tension, τ. Such a penalized formulation leads to a problem that only involves the velocity field. Therefore, the aim is to clarify how the constitutive equation is modified when introducing two penalty parameters. For that purpose we consider:

$$\nabla \cdot \mathbf{v} + \lambda p = \mathbf{D} : \mathbf{I} + \lambda p = 0, \tag{5.89}$$

and

$$\mathbf{D} : \mathbf{a} - \epsilon \tau = 0. \tag{5.90}$$

Where the coefficients λ and ϵ are chosen to ensure numerically incompressibility and fiber inextensibility. Both constraints are ensured as long as both penalty coefficients are small enough. Isolating p and τ from Eqs. (5.89) and (5.90), it results

$$p = -\frac{1}{\lambda} \mathbf{D} : \mathbf{I}, \tag{5.91}$$

and

$$\tau = \frac{1}{\epsilon} \mathbf{D} : \mathbf{a}. \tag{5.92}$$

If both pressure and fiber tension are penalized the constitutive equation reduces to,

$$\boldsymbol{\sigma} = \frac{1}{\lambda} (\mathbf{I} \otimes \mathbf{I}) : \mathbf{D} + \frac{1}{\epsilon} (\mathbf{a} \otimes \mathbf{a}) : \mathbf{D} + 2\eta \mathbf{D}, \tag{5.93}$$

where a very high effective viscosity acts along the fiber direction. This formulation was intensively considered in [Ghnatios *et al.* 2016] where a variety of results were presented and discussed, proving the potentiality of the approach. However neither fiber tension nor the pressure field were calculated because the penalty formulation does not allow an easy and accurate post-calculation from the calculated velocity field.
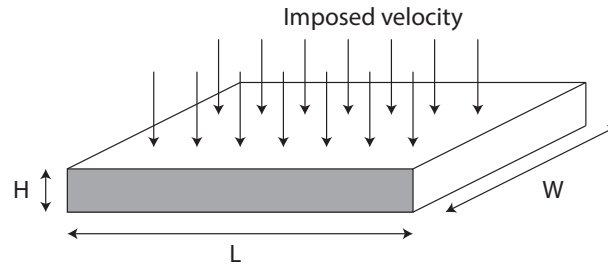
Figure 5.31 – Laminate geometry during compression molding of the laminate under prescribed velocity.

5.3.4 Numerical results

The numerical results discussed hereafter consider several cases starting from Stokes problem, then a single ply occupying the whole gap, then considering laminates composed of two plies with different relative orientations.

The main aim of this section is to show that there exists an in-plane-out-of-plane separated representation when both the pressure and the fiber tension have been introduced as Lagrange multipliers. Therefore, 3D finite element solvers for Stokes and Ericksen problems have been developed in order to check the accuracy of the separated representation solutions. The 3D-FEM solution will be taken as reference for validating the one involving a separated representation 3D-SR calculated within the PGD framework.

The domain occupied by the laminate has length L, width W and thickness H, i.e. $\Xi = \Omega \times \mathcal{I}$, with $\Omega = [0, L] \times [0, W]$ and $\mathcal{I} = [0, H]$, with $\mathbf{x} \in \Omega$ and $z \in \mathcal{I}$.

It is assumed that during compression, the upper wall moves down with a prescribed velocity V during the consolidation. Thus, mass conservation leads to significant velocity variations within Ω, e.g. the central point has a null in-plane velocity because of the symmetry condition whereas the in-plane velocity is maximal at the laminate lateral boundaries $\partial\Omega \times \mathcal{I}$. Moreover, when taking into account the through-thickness complex kinematics in multiaxial laminates, a sufficiently detailed solution is also required along the thickness direction to capture all its richness. Figure 5.31 depicts the laminate geometry as well as the squeezing conditions.

Numerical results will be presented sometimes on the middle plane $z = H/2$, sometimes along the thickness at the intersection line between planes $x = a$ and $y = b$, with $a \in [0, L]$ and $b \in [0, W]$.

5.3.4.1 Stokes flow

First the Stokes problem solution is addressed. Numerical issues as the ones related to the enforcement of the so-called LBB stability condition in the 3D-SR (in-plane-out-of-plane separated representation) will be deeply discussed. Viscosity is set to 1000 Pa · s and the compression velocity to $V = -1$ m · s^{-1}. Two domains were addressed, the first perfectly cubic of length 1 m. The second flow problem was defined in the narrow gap of dimensions $L \times W \times H = 0.5 \times 0.5 \times 0.001$ (all the lengths in meters) with a prescribed compression velocity of $V = -0.1$ mm · s^{-1}. In this case the 3D flow problem solution could be approximated using the lubrication theory.
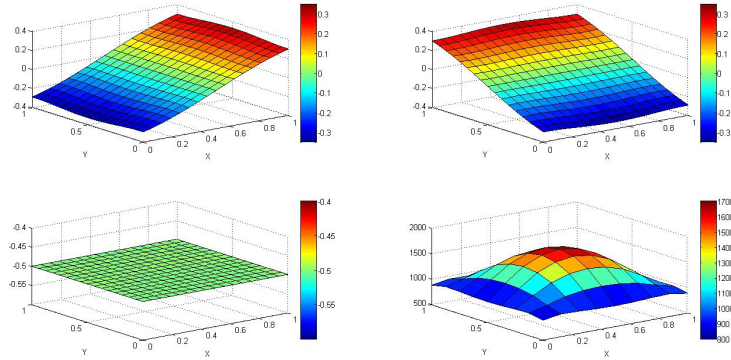
Figure 5.32 – $u(\mathbf{x}, z = H/2)$ (top-left), $v(\mathbf{x}, z = H/2)$ (top-right), $w(\mathbf{x}, z = H/2)$
(bottom-left), and $p(\mathbf{x}, z = H/2)$ (bottom-right), associated with the solution of the
Stokes problem using a stable 3D-FEM discretization.

3D-FEM A standard stable 3D finite element discretization has been used for solving
the Stokes problem in the cubic domain of unit size ($L = W = H = 1$). The 3D-FEM
solution will be taken as reference for evaluating the 3D solution making use of an in-plane-
out-of-plane separated representation. A stable Q2/P1 is considered for discretizing the 3D
mixed formulation within the FEM framework.

Figure 5.32 depicts the three velocity components and the pressure field on the middle
plane $z = H/2$. Because the problem symmetry the velocity components $u(\mathbf{x}, z)$ and
$v(\mathbf{x}, z)$ vanish at $x = L/2$ and $y = W/2$ respectively. Moreover the component w remains,
as expected, almost constant on the middle plane $z = H/2$. The pressure field exhibits a
global maximum in the center of the middle plane of about 1669 Pa.

Figure 5.33 shows the different components of the velocity field as well as the pressure
along the line defined as the intersection between planes $x = 0.625$ and $y = 0.625$. It can be
noticed that boundary conditions are satisfied, with velocities u and v vanishing at $z = 0$
and $z = H$, whereas velocity w vanishes at $z = 0$ and corresponds to the compression
velocity at $z = H$. Velocities u and v exhibit a parabolic profile along the thickness whose
maximum is located at the middle plane $z = H/2$. Velocity w exhibit an almost cubic
evolution along the thickness. The pressure field presents a minimum in the middle plane
being maximum when approaching the upper and bottom walls.

3D-SR Now, the same problem is solved using a separated representation of both the
velocity and pressure functions within the PGD framework. The in-plane functions were
discretized using 2D Q2/P1 approximations for the velocity and pressure fields respectively,
whereas a 1D Q2/P1 was considered for approximating the velocity and pressure functions
depending on the thickness coordinate respectively.

Figure 5.34 depicts the reconstructed velocity and pressure fields on the middle plane
$z = H/2$. In the approximation of functions depending on the in-plane coordinates and
those depending on the one related to the domain thickness, we considered meshes equiv-
alent to the one considered in the 3D-FEM solution. Results are in perfect agreement to
those obtained using the 3D-FEM. Despite the very coarse mesh considered the maximum
gap in the pressure field was lower than 0.5%.

Figure 5.33 – $u(\mathbf{x} = (0.625, 0.625)^T, z)$ (top-left), $v(\mathbf{x} = (0.625, 0.625)^T, z)$ (top-right), $w(\mathbf{x} = (0.625, 0.625)^T, z)$ (bottom-left), and $p(\mathbf{x} = (0.625, 0.625)^T, z)$ (bottom-right), associated with the solution of the Stokes problem using a stable 3D-FEM discretization..

Figure 5.34 – $u(\mathbf{x}, z = H/2)$ (top-left), $v(\mathbf{x}, z = H/2)$ (top-right), $w(\mathbf{x}, z = H/2)$ (bottom-left), and $p(\mathbf{x}, z = H/2)$ (bottom-right), associated with the solution of the Stokes problem using the in-plane-out-of plane separated representation.
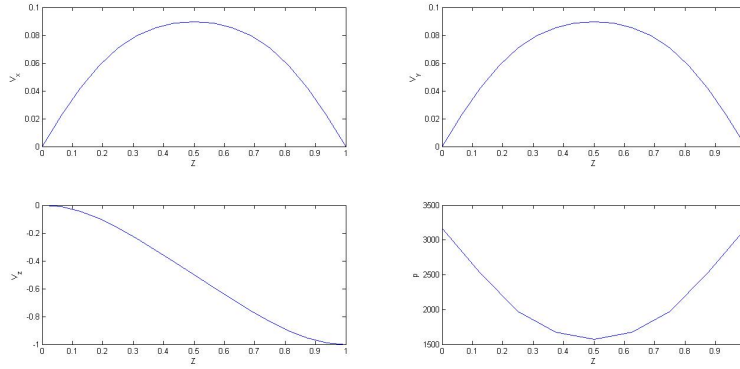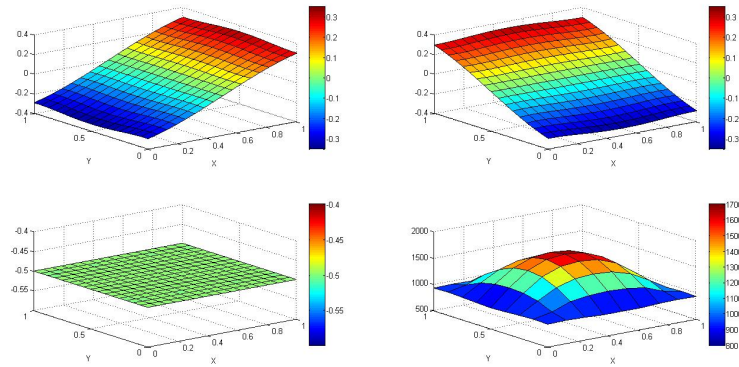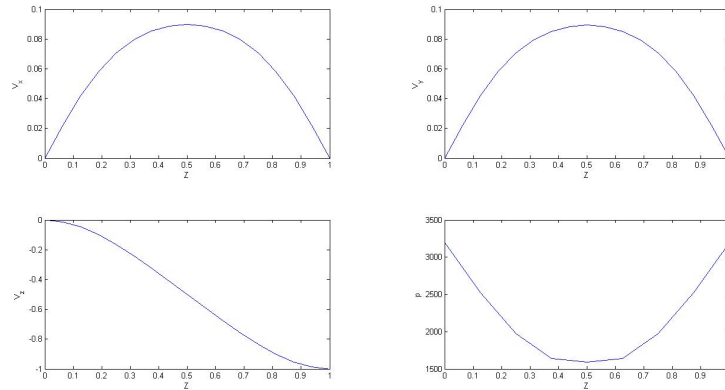
Figure 5.35 – $u(\mathbf{x} = (0.625, 0.625)^T, z)$ (top-left), $v(\mathbf{x} = (0.625, 0.625)^T, z)$ (top-right), $w(\mathbf{x} = (0.625, 0.625)^T, z)$ (bottom-left), and $p(\mathbf{x} = (0.625, 0.625)^T, z)$ (bottom-right), associated with the solution of the Stokes problem using the in-plane-out-of plane separated representation.

Figure 5.35 depicts the velocity and pressure profiles along the domain thickness at position $x = y = 0.625$. Again the solution obtained within the PGD framework perfectly agrees with the reference solution obtained from the stable 3D finite element discretization previously presented.

Moreover, in order to check the stability conditions (LBB) different choices were considered. First we considered Q2/Q2 approximations in the plane (that in 2D does not fulfill LBB stability conditions) whereas the one considered for the problem defined in the thickness was assured stable (Q2/P1). It can be noticed in figure 5.36 that the resulting in-plane-out-of-plane approximation is not stable, and that the characteristic oscillations appear in the in-plane solution (in which stability fails).

We also check another approximation expected violating the LBB stability conditions, the one using a Q2/P1 approximation in the plane for velocities and pressure respectively and Q2/Q2 for functions depending on the thickness. The last is expected violating the LBB stability conditions. Fig. 5.37 exhibits oscillations precisely in the pressure field along the thickness direction, that can be attributed to the wrong approximation choice for the functions depending of the thickness coordinate and involved in the velocity and pressure representation.

A first conclusion stressed from these numerical experiments is that when using in the separated representation approximations whose tensor product corresponds to 3D stable approximations (i.e. fulfilling the LBB condition) those separated approximations remain stable.

3D-SR solution in a narrow gap The flow problem is now defined in a very narrow gap because is in these circumstances that the use of separated representations could be extremely advantageous. As previously indicated now the domain $L \times W \times H = 0.5 \times 0.5 \times 0.001$ (all the units in meters) being again the viscosity $\eta = 1000 \, \mathrm{Pa \cdot s}$ and the compression velocity applied at the upper wall $V = -0.1 \, \mathrm{mm \cdot s^{-1}}$.

Figures 5.38 and 5.39 depict velocities and pressure fields obtained using the in-plane-
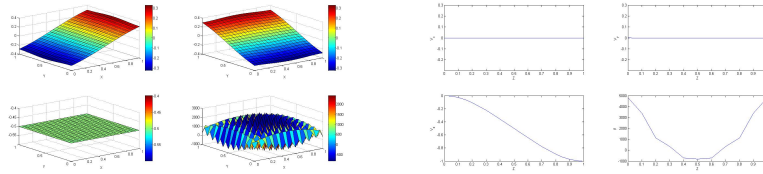
Figure 5.36 – Velocity a pressure fields on the middle plane $z = 0.5$ (left) and solution along the thickness ($\mathbf{x} = (0.625, 0.625)^T, z$) when using the in-plane-out-of-plane separated representation with in-plane and thickness approximations Q2/Q2 and Q2/P1 respectively, for the functions involved in the velocity and pressure representation.
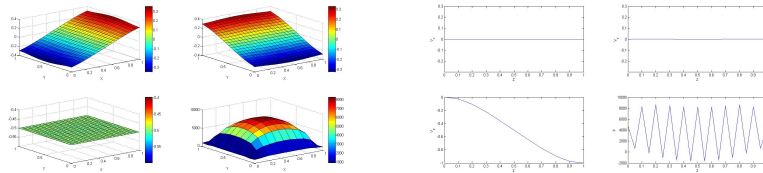
Figure 5.37 – Velocity a pressure fields on the middle plane $z = 0.5$ (left) and solution along the thickness ($\mathbf{x} = (0.625, 0.625)^T, z$) when using the in-plane-out-of-plane separated representation with in-plane and thickness approximations Q2/P1 and Q2/Q2 respectively, for the functions involved in the velocity and pressure representation.
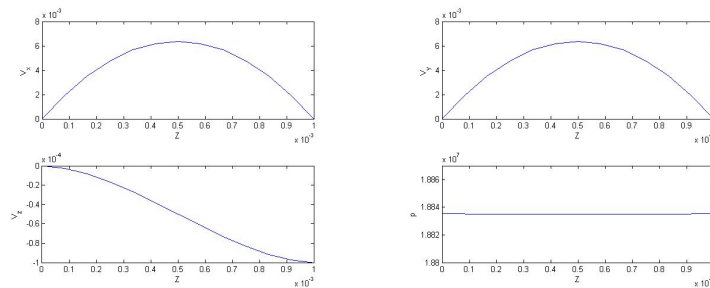
Figure 5.38 – $u(\mathbf{x} = (0.33, 0.333)^T, z)$ (top-left), $v(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-right), $w(\mathbf{x} = (0.33, 0.33)^T, z)$ (bottom-left), and $p(\mathbf{x} = (0.33, 0.33)^T, z)$ (bottom-right), associated with the solution of the Stokes problem in a narrow-gap using the in-plane-out-of plane separated representation.
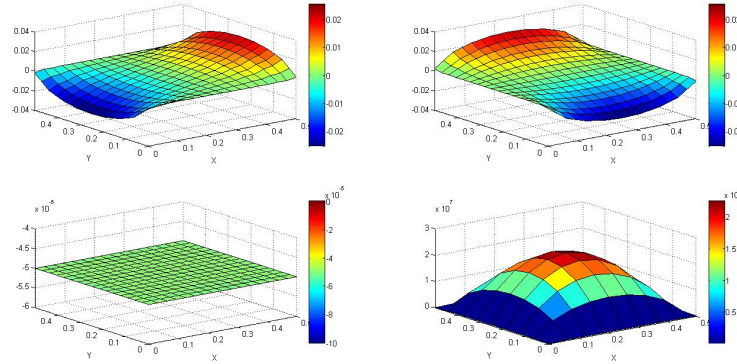
Figure 5.39 – $u(\mathbf{x}, z = H/2)$ (top-left), $v(\mathbf{x}, z = H/2)$ (top-right), $w(\mathbf{x}, z = H/2)$ (bottom-left), and $p(\mathbf{x}, z = H/2)$ (bottom-right), associated with the solution of the Stokes problem in a narrow-gap using the in-plane-out-of plane separated representation.

pot-of-plane separated representation in conjunction with stable approximations (Q2/P1 in the plane and also Q2/P1 in the thickness). It is important to note that as expected from the lubrication theory pressure becomes constant all along the thickness.

5.3.4.2 Laminate composed of a single Ericksen ply.

The first test case consists of a laminate composed of a single ply described by the Ericksen constitutive equation, with the unidirectional continuous fiber reinforcement oriented along the x-coordinate axis. Thus, the reinforcement orientation is defined by $\mathbf{p}^T = (1, 0, 0)$, implying the orientation tensor

$$\mathbf{a} = \left(\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right). \tag{5.94}$$

The squeeze flow takes place within the narrow gap $L = W = 0.5$ and $H = 10^{-3}$ (units in meters) being the fluid viscosities $\eta_L = 100$ Pa·s and $\eta_T = 100$ Pa·s. The squeezing rate was again $V = -0.1$ mm·s^{-1}.

3D-FEM First it is solved the flow problem by using a stable 3D finite element discretization (Q2/P1/P1 for the velocity, pressure and tension respectively). The finite element solution will be considered as the reference one for checking the solutions obtained within the separated representation (PGD) framework.

Figure 5.40 depicts all the unknown fields, the three velocity components (u, v, w), the pressure p and the tension τ. As expected and because the symmetry, the velocity in the fibers direction vanishes (a constant value is not possible because the flow problem symmetry and on the other hand non constant velocities will imply extensibility that is not allowed in the Ericksen fluid model). Component $w(\mathbf{x}, z = H/2)$ is as expected constant and $v(\mathbf{x}, z = H/2)$ has a linear variation vanishing at $y = W/2$ because the flow problem symmetry. Both the pressure and the tension exhibit a parabolic profile, the first expected

Figure 5.40 – $u(\mathbf{x}, z = H/2)$ (top-left), $v(\mathbf{x}, z = H/2)$ (top-right), $w(\mathbf{x}, z = H/2)$ (middle-left), $p(\mathbf{x}, z = H/2)$ (middle-right) and $\tau(\mathbf{x}, z = H/2)$ (bottom) associated with the solution of the Ericksen fluid flow problem in a narrow-gap using a 3D stable finite element discretization.

from the lubrication theory and the second quite intuitive because at $y = W/2$ the fibers resists a flow that in their absence will take place in the x-direction, and decrease until vanishing at $y = 0$ and $y = W$. Figure 5.41 shows the velocity, pressure and tension profiles along the gap thickness at position $\mathbf{x} = (0.312, 0.312)^T$, that exhibit the expected behavior.

We proved from our numerical experiments that richer tension approximation do not satisfy the LBB stability conditions. Thus, approximating the tension in the same space than the pressure seems a safe choice.

3D-SR Again an in-plane-out-of-plane separated representation of velocities, pressure and fiber tension was considered within the PGD framework. However, in the case of a single fluid layer, the Lagrange multiplier associated with the fiber tension is not coupled along the thickness direction, i.e. the tension at different z-coordinates are fully decoupled. Thus, no equation is found in order to compute the functions $T_i^\tau(z)$ and for this reason we considered the simplest choice of assuming they are the same that the ones used in the pressure separated representation, i.e.

$$p(\mathbf{x}, z) = \sum_{i=1}^{N} P_i^p(\mathbf{x}) \cdot T_i^p(z), \qquad (5.95)$$

and

$$\tau(\mathbf{x}, z) = \sum_{i=1}^{N} P_i^\tau(\mathbf{x}) \cdot T_i^p(z), \qquad (5.96)$$

Figure 5.42 and 5.43 presents similar results that the ones depicted in Figs. 5.40 and 5.41 when the 3D mixed formulation is solved by using a stable in-plane-out-of-plane separated representation within the PGD framework, considering Q2/P1/P1 approximations for the functions depending on the in-plane coordinates and also Q2/P1/P1 for those depending on the z-coordinate. The obtained results are almost identical the the ones obtained by using the more experienced 3D finite element discretization discussed in the previous section.

Figure 5.41 – $u(\mathbf{x} = (0.312, 0.312)^T, z)$ (top-left), $v(\mathbf{x} = (0.312, 0.12)^T, z)$ (top-right), $w(\mathbf{x} = (0.312, 0.312)^T, z)$ (middle-left), $p(\mathbf{x} = (0.312, 0.312)^T, z)$ (middle-right) and $\tau(\mathbf{x} = (0.312, 0.312)^T, z)$ (bottom), associated with the solution of the Ericksen fluid flow problem in a narrow-gap using a 3D stable finite element discretization.
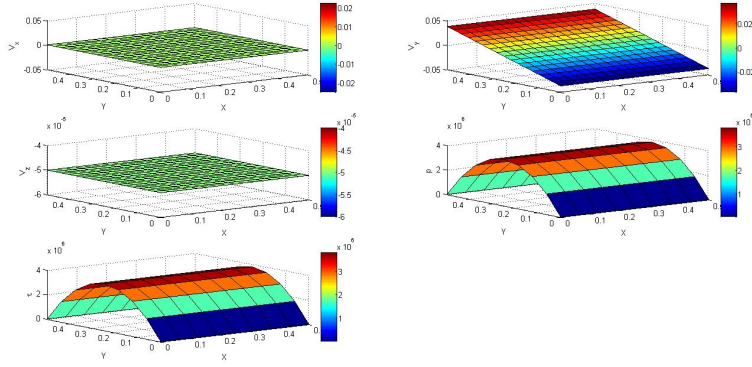

Figure 5.42 – $u(\mathbf{x}, z = H/2)$ (top-left), $v(\mathbf{x}, z = H/2)$ (top-right), $w(\mathbf{x}, z = H/2)$ (middle-left), $p(\mathbf{x}, z = H/2)$ (middle-right) and $\tau(\mathbf{x}, z = H/2)$ (bottom) associated with the solution of the Ericksen fluid flow problem in a narrow-gap using a 3D in-plane-out-of-plane separated representation.
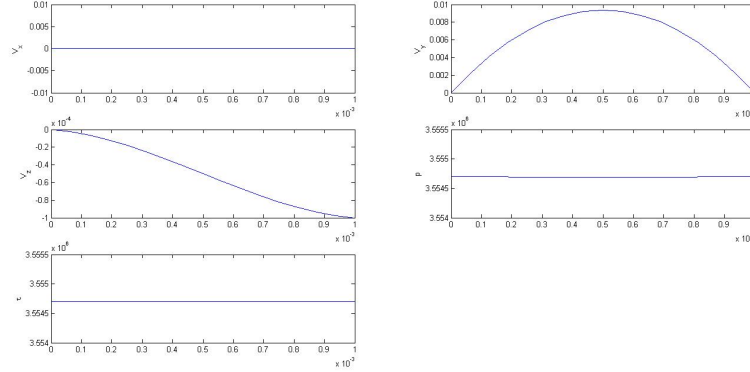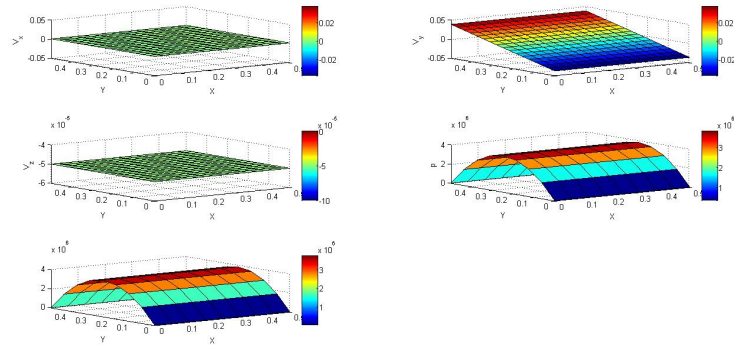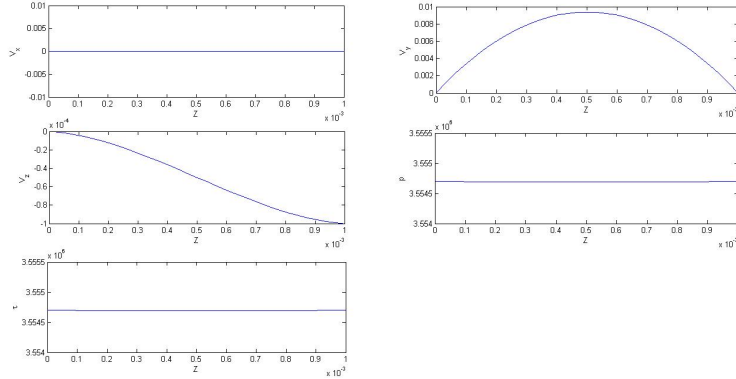
Figure 5.43 – $u(\mathbf{x} = (0.312, 0.312)^T, z)$ (top-left), $v(\mathbf{x} = (0.312, 0.12)^T, z)$ (top-right), $w(\mathbf{x} = (0.312, 0.312)^T, z)$ (middle-left), $p(\mathbf{x} = (0.312, 0.312)^T, z)$ (middle-right) and $\tau(\mathbf{x} = (0.312, 0.312)^T, z)$ (bottom), associated with the solution of the Ericksen fluid flow problem in a narrow-gap using a 3D in-plane-out-of-plane separated representation.

5.3.4.3 Laminate composed of two Ericksen plies.

The following test case consists of a laminate composed of 2 plies described by Ericksen constitutive equation. The ply dimensions are again $L \times W \times H = 0.5 \times 0.5 \times 10^{-3}$ (all units in meters) with a compression velocity applied at the upper wall $V = -0.1$ mm $\cdot$ s^{-1}, with the same viscosities that were employed previously. The fiber orientation in the bottom ply was given by $\mathbf{p}^B = (1, 0, 0)^T$ whereas in the upper ply they were oriented along the y-direction, i.e. $\mathbf{p}^U = (0, 1, 0)^T$.

The discretization was carried out again using a stable 3D finite element approximation (Q2/P1/P1). The solution (velocity, pressure and tension) profiles though the thickness at position $\mathbf{x} = (0.33, 0.33)^T$ are depicted in Fig. 5.44. As expected we obtain two parabolic profiles for the velocities u and v, the first vanishing in the bottom ply (because the fiber inextensibility and the flow problem symmetry) and exhibiting a parabolic profile in the upper ply; and the symmetric behavior for the velocity v. As expected the velocity component w evolves smoothly, the pressure remains almost constant, however, a peak in the tension is noticed at the plies interface.

The origin of the tension peak is easy to understand. The parabolic profile of $u(\mathbf{x}, z)$ through the ply thickness z in the upper ply implies a shear rate and consequently a shear stress at the interface. However, at the interface the x-component of the traction $\mathbf{T} = \boldsymbol{\sigma} \cdot \mathbf{e}_z$ ($\mathbf{e}_z$ being the unit vector defining the z-coordinate axis) computed at the bottom ply for equilibrating the shear stress associated to the parabolic profile in the upper-ply implies a non-null component xz of the rate of strain tensor, i.e. $\mathbf{D}_{xz} \neq 0$ (note that fiber tension τ is not involved in the expression of traction $\mathbf{T}$). Thus the u velocity in the bottom ply cannot be exactly zero, there is a boundary layer located at the interface in which it activates the fiber tension. Because as just indicated the fibers tension do not communicate along the thickness, the tension singularity remains located at the interface level and does not propagate in the bottom ply. The same reasoning applies for the parabolic profile of $v(\mathbf{x}, z)$ in the bottom ply that implies a tension singularity in the upper ply at the interface

Figure 5.44 – $u(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-left), $v(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-right), $w(\mathbf{x} = (0.33, 0.33)^T, z)$ (middle-left), $p(\mathbf{x} = (0.33, 0.333)^T, z)$ (middle-right) and $\tau(\mathbf{x} = (0.33, 0.33)^T, z)$ (bottom), associated with the solution of the two ply Ericksen fluid flow problem in a narrow-gap using a stable 3D finite element discretization.

neighborhood. By diminishing the viscosity the shear stress decreases and then the tension peaks. This tendency has been verified numerically.

Solving the same problem by using an in-plane-out-of-plane separated representation seems a tricky issue because the inevitable singularity that the Ericksen model induces at the plies interface when the orientation of fibers evolves from one ply to its contiguous one. Obviously we proved that if the different plies are oriented along the same direction no peak appears in the solution and the the solution. The resulting solution in that case results the one associated with a single ply having the laminate thickness.

The solution for the considered case of two plies with different orientations is depicted in Fig. 5.45, that reveals all the expected tendencies, in particular the peak in the tension that now spreads a little bit more from the interface.

Due to these tension peaks when proceeding within the PGD framework the pressure solution is slightly polluted, revealing an almost constant value across the gap thickness a bit lower that the one obtained when using the 3D finite element discretization. For this reason we finally decided to consider a velocity-pressure mixed formulation whereas the tension is treated from a penalty formulation. In fact accurate pressure solutions are required in order to characterize the laminate behavior, however tension is only required for evaluating the defect risks related for example with compressive fiber tensions. Thus one could imagine that tension could be reconstructed from the velocity-pressure solution, and even if its accuracy is compromised it suffices for the purpose of evaluating defect risks. Figure 5.46 depicts the velocity and pressure solution where higher accuracy is noticed concerning the pressure field.

5.3.4.4 Rheological characterization

In this section we address the rheological characterization of a laminate composed of the two Ericksen plies considered in the previous section. We consider different squeezing rates and for each one after calculating the velocity and pressure fields (from the stabilized and tension penalized in-plane-out-of-plane separated representation), tension $\boldsymbol{\sigma} \cdot \mathbf{e}_z$ is calculated
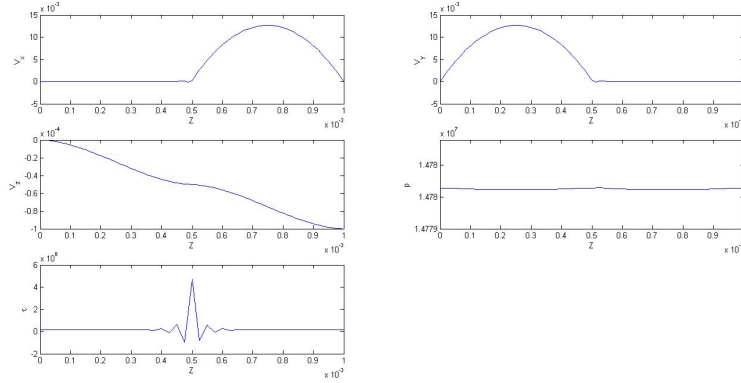
Figure 5.45 – $u(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-left), $v(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-right), $w(\mathbf{x} = (0.33, 0.33)^T, z)$ (middle-left), $p(\mathbf{x} = (0.33, 0.333)^T, z)$ (middle-right) and $\tau(\mathbf{x} = (0.33, 0.33)^T, z)$ (bottom), associated with the solution of the Ericksen fluid flow problem in a narrow-gap using an in-plane-out-of-plane separated representation.
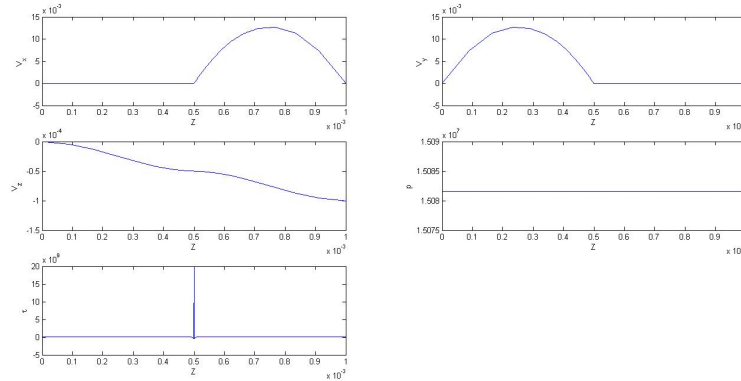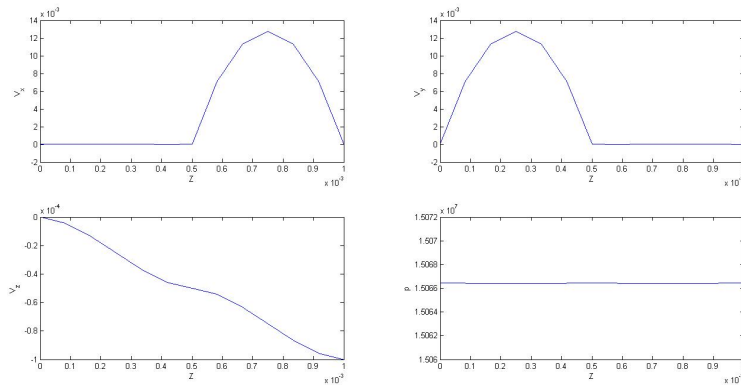

Figure 5.46 – $u(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-left), $v(\mathbf{x} = (0.33, 0.33)^T, z)$ (top-right), $w(\mathbf{x} = (0.33, 0.33)^T, z)$ (bottom-left) and $p(\mathbf{x} = (0.33, 0.333)^T, z)$ (bottom-right) associated with the solution of the Ericksen fluid flow problem in a narrow-gap using an in-plane-out-of-plane separated representation with a penalized tension.
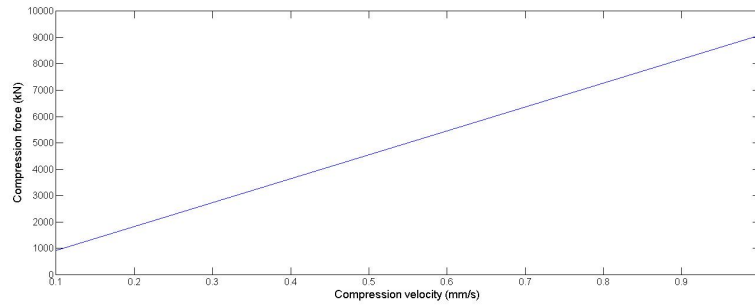
Figure 5.47 – Applied compression force versus squeeze rate for the two plies Ericksen laminate discussed in the previous section.

on the upper plate and then the resultant of its normal component obtained. Figure 5.47 depicts the force/squeeze rate behavior, that as expected evolves linearly and from which one could extract an equivalent newtonian viscosity.

5.3.5 Conclusions

The in-plane-out-of-plane separated representation has been succesfully applied to simulate the squeeze flow of multiaxial laminates, able to present resolution levels never envisaged until now. This particular separated representation allows calculating extremely detailed 3D solutions while keeping the computational complexity characteristic of 2D problems. Thus, extremely high resolutions can be attained in the thickness direction, able to capture localized behaviors.

In this work we succeeded to solve mixed velocity-pressure-tension formulations within an in-plane-out-of-plane separated representation. In order to address multiaxial laminates involving tension localized behaviors, we proposed a mixed velocity-pressure with a penalized tension, that allowed very accurate velocity and pressure solutions, and a reasonable reconstructed tension, accurate enough for prediction defect risks.

Conclusions and Perspectives

This thesis was devoted to the strengthen the link between data and standard simulation based numerical methods. The applications presented in this manuscript point in the direction of Data Driven Dynamic Application Systems (DDDAS), where the usage of data to correct existing models becomes of crucial interest.

The purpose of this thesis was fourfold.

Data-Driven Techniques

In chapter 2, we developed several techniques within the data-driven framework. In section 2.1, a numerical technique able to use a data set substituting an explicit constitutive equation in the solid mechanic framework was presented. By doing that, constitutive error modeling is circumvented, or in any case alleviated, as soon as the measured data set is free of error. This numerical technique is merged consistently with finite elements making its use possible in any finite element code.

Section 2.2 explores one of the current drawbacks of data-driven techniques, the data generation. Even though it seems contradictory, the truth is that data is not so abundant when considering nowadays testing facilities. For instance, most stress-strain curves can only be extracted under certain homogeneity hypothesis i.e. uniaxial or biaxial loading. As a consequence, reaching complex stress-strain states becomes a cumbersome task. This section presents a novel technique able to extract complex stress-strain states from an non-homogeneous experiment. Indeed, this inverse procedure is in perfect harmony with the numerical technique presented in section 2.1 where diverse stress-strain states were mandatory.

Section 2.3 tackles another data-driven application upscaling microscopic behaviours to the macro scale. In this case, we proposed a novel upscaling technique where microscopic effects were collected in a data set (or constitutive manifold). This data set can be ultimately used to solve the macroscopic solution. More in detail, a microscopic short fiber suspension fluid in porous media is successfully upscaled to engendrate an effective Darcy's law.

It is important to emphasize that these promising results are really the early stages of a new research topic. Important steps, regarding noisy data or data generation, still need to be overcome. Being able to apply such inverse constitutive identification procedure within the inelastic framework (i.e. plasticity or damage) constitute an active line of research. Indeed, the identification of internal state variables (i.e. cumulated plastic strain or damage state) adds complexity to the inverse procedure. Moreover, inverse constitutive parameter identification requires an extra-effort to guarantee certain thermodynamic laws. For instance, the numerical method should be able to avoid non-physical parameter such as negative Young's modulus or non-symmetric stress states.

Data Completion Techniques

In chapter 3 we explored different data completion techniques. These techniques play a major role within the data-driven framework as soon as a low amount of data is available. Hence, such data completion techniques are very helpful to estimate information at query points where data was not available.

Section 3.1 introduces a novel data completion technique, the sparse Proper Generalized Decomposition. It combines both the separation of variables to circumvent the curse of dimensionality together with a collocation procedure to use data only wherever it is available. This data completion algorithm has been succesfully applied in three different applications shown in section 3.2.

Section 3.3 applies the concept of sparse identification techniques [Brunton *et al.* 2016] to several computational mechanics applications such as parametric response surface reconstruction, transient problems or constitutive manifold completion.

Even though these data completion methodologies have been successfully applied to a vast variety of problems. It is important to remark that the choice of the basis plays a fundamental role to guarantee the efficiency of the method. For instance, the way the basis is adapted for different number of modes in the sPGD framework affects the inference properties of the method. Similarly, sparse identification techniques relies on the hypothesis that few coefficients are required to represent a given response. Therefore, the basis should be chosen in a way that this hypothesis is fulfilled as much as possible. Having said that, developing efficient error indicators allowing to adapt the basis online could constitute a line of research.

Multi PGD based on the Partition of Unity Method

As stated in chapter 4, Model Order Reduction (MOR) techniques could be seen as natural ally to achieve dynamic data-driven corrections. However, MOR methodologies present several difficulties when facing highly non-linear solutions. Section 4.1 introduces a new numerical method that combines standard PGD rationale together with a macro partition of unity allowing to combine different PGDs in the same domain. By doing that, each individual PGD handles a rather local domain where the solution tends to be more linear, improving ultimately the separability of each PGD.

Section 4.2 extends the multi-PGD methodology to treat multi-scale problems. The consistent overlap between different subdomains automaticly solves the transition between different scales.

In the light of the results, the multi-PGD framework enhances the separability of the PGD methodology requiring less number of modes to represent the solution up to a certain accuracy. However, important steps, considering a macro mesh refinement in areas where the residual is high just like high order macro partition of unity, have to be considered to enhance the performance of the method.

Applications of PGD In-Plane-Out-Of-Plane Separations

Several applications of the PGD In-Plane-Out-Of-Plane Separated Representations are treated in Chapter 5. This kind of separated representation has been successfully applied in many mechanical systems and complex structures involving plate and shell parts whose main particularity is having a characteristic dimension (the one related to the thickness) much lower that the other ones (in-plane dimensions) [Bognet *et al.* 2012, Bognet *et al.* 2014]. The electromagnetic response of thin composite plies is studied in section 5.1. A numerical algorithm able to efficiently compute the electromagnetic field is such degenerated domain is of crucial interest to understand the heating process using microwaves. Section 5.2 introduces a new hybrid explicit/implicit in-plane-out-of-plane separated representation for dynamic problems defined in plate-like domains computing efficiently 3D solutions and where the stability constraints are exclusively determined by the coarser in-plane discretizations. All operators involving the out-of-plane coordinate are treated online compensating

for the unaffordable time step associated to this dimension.

Section 5.3 explores the kinematic of thermoplastic composite plies. Particularly, the material is modeled by means of an anisotropic fluid which is incompressible and inextensible. Questions such as the stability of the separated representation when facing a saddle-point problem are considered.

To sum up, we strongly believe that this manuscript constitutes a early stage towards the fulfillment of the so-called Dynamic Data Driven Application Systems. As it can be appreciated in the individual conclusions of each section, the numerical techniques proposed in this work might open the route to new research lines.

A simple microstructural viscoelastic model for flowing foams

The work presented herein makes reference to a published paper:

- R. Ibáñez, A. Scheuer, E. Abisset-Chavanne, F. Chinesta, A. Huerta, *A simple microstructural viscoelastic model for flowing foams*. International Journal of Material Forming, 1-12, 2018.

A.1 Introduction

Aqueous foams are concentrated dispersions of gas bubbles in a surfactant solution. Their structures are organized over a large range of length scales and complex flows take place at different scales [S. Cohen-Addad 2013].

The proposal of macroscopic constitutive equations allows for the efficient modelling and simulation of industrial processes involving the flow of foams [Cheddadi *et al.* 2008, Benito *et al.* 2008]. Usually, such descriptions remain however too phenomenological, and even though they predict accurately the flow kinematics, microstructure information remains often unaccessible. On the opposite side, fully microscopic simulations allow for very detailed descriptions of the foam microstructural evolution [Bikard *et al.* 2005]. However such approaches fail to address scenarios of industrial interest that usually involve the flow of foams in very large and complex 3D geometries.

The macroscopic flow model is expected to depend on the cellular structure: cell size, shape and orientation, as well as on the fluid rheology and the surface tension. Moreover, cell shape and orientation are induced by the flow. This microscopic information could be introduced into a macroscopic flow model by using standard upscaling and homogenization techniques. Thus inspired by [Feyel 2003], at some locations in the domain in which an effective homogeneous fluid flows, we could attach a representative volume containing several cells, whose size, shape and orientation depend on the considered location. Now, a detailed microscopic calculation could be carried out in order to determine the effective fluid rheology. However, such a route, widely and successfully considered in a variety of fields, remains expensive from a computational point of view despite some attempts at combining it with advanced model reduction techniques [Lamari *et al.* 2010].

The most appealing description consists of a macroscopic flow model making use of some conformational variables describing the main microstructural features, as widely considered in the field of multiscale polymer modelling [Keunings 2004, Binetruy *et al.* 2015]. Thus, in [Karimi *et al.* 2017] the authors study flows during foaming, considering cell evolutions but without addressing the shape orientation and then, the induced anisotropy. Richer microstructure descriptions can be obtained by using a set of configurational coordinates, from which a conformation tensor can be derived and a macroscopic constitutive equation

established. This route was successfully considered in [Tlili *et al.* 2015]. In the present work, we propose an alternative simpler microstructure description and its coupling with the macroscopic flow.

Remark. In the sequel, we consider the following tensor products, where Einstein's summation convention is assumed:

- if **a** and **b** are first-order tensors, the single contraction · reads $(\mathbf{a} \cdot \mathbf{b}) = a_j \, b_j$;

- if **a** and **b** are first-order tensors, the dyadic product $\otimes$ reads $(\mathbf{a} \otimes \mathbf{b})_{jk} = a_j \, b_k$;

- if **a** and **b** are respectively second and first-order tensors, the single contraction · reads $(\mathbf{a} \cdot \mathbf{b})_j = a_{jm} \, b_m$;

- if **a** and **b** are second-order tensors, the single contraction · reads $(\mathbf{a} \cdot \mathbf{b})_{jk} = a_{jm} \, b_{mk}$;

- if **a** and **b** are second-order tensors, the double contraction : reads $(\mathbf{a} : \mathbf{b}) = a_{jk} \, b_{kj}$.

A.2 Cell conformation

A very simple description of a 3D cell consists of a deformable ellipsoid (with constant volume) ranging from the spherical shape to the infinite aspect ratio ellipsoid (rod). Such an ellipsoid could be represented by means of three orthogonal extensible springs, with reference length $2L^0$ and stiffness $\mathcal{K}$. In the sequel, we restrict our analysis to 2D scenarios (by considering ellipses instead of ellipsoids) but the derived models and their numerical solution procedures can be straightforwardly extended to 3D.

First, we consider the kinematics of a single linear elastic dumbbell as starting point for elaborating the cell conformation.

A.2.1 Kinematics of an extensible rod

The extensible rod, of reference length $2L^0$ and assumed aligned in direction **p** (**p** having a unit norm), is represented by an elastic spring of length $2L$ (in the deformed state) and stiffness $\mathcal{K}$ equipped with two beads at its extremities where hydrodynamic forces act. In the sequel, the word hydrodynamic refers to the viscous drag force and not to the one considered in other works to describe the effects of a bead kinematics on the others from the use of the Oseen tensor. These forces scale with the fluid - bead relative velocity, the former given by $\mathbf{v}_0 + \nabla\mathbf{v} \cdot \mathbf{p}L$ and the latter by $\mathbf{v}_G + \dot{\mathbf{p}}L + \mathbf{p}\dot{L}$, where $\mathbf{v}_0$ is the unperturbed fluid velocity at the rod center of gravity and $\mathbf{v}_G$ the velocity of the rod centre of gravity. A sketch of the rod and the forces acting on it is depicted in Fig. A.1.

The system is assumed inertialess, that implies the equilibrium of forces and torques. The first implies $\mathbf{F}^H(\mathbf{p}L) + \mathbf{F}^H(-\mathbf{p}L) = \mathbf{0}$, leading to $\mathbf{v}_0 = \mathbf{v}_G$, that is, the rod centre of gravity moves with the fluid.

Now, to prevent a resultant torque, force $\mathbf{F}^H(\mathbf{p}L)$ must align with **p**, i.e. $\mathbf{F}^H(\mathbf{p}L) = \lambda\mathbf{p}$, $\lambda \in \mathbb{R}$. Thus, we have

$$\mathbf{F}^H(\mathbf{p}L) = \xi(\nabla\mathbf{v} \cdot \mathbf{p}L - \dot{\mathbf{p}}L - \mathbf{p}\dot{L}) = \lambda\mathbf{p}, \tag{A.1}$$

that multiplying by **p** and taking into account that $\mathbf{p} \cdot \mathbf{p} = 1$ and consequently $\mathbf{p} \cdot \dot{\mathbf{p}} = 0$, yields

$$\xi(\nabla\mathbf{v} : (\mathbf{p} \otimes \mathbf{p})L - \dot{L}) = \lambda, \tag{A.2}$$

$$\mathbf{F}^H(\mathbf{p}L) = \xi(\mathbf{v}_0 + \nabla\mathbf{v}\cdot\mathbf{p}L - \mathbf{v}_G - \dot{\mathbf{p}}L - \mathbf{p}\dot{L})$$

$2L$

$\mathbf{p}$

G

$$\mathbf{F}^H(-\mathbf{p}L) = \xi(\mathbf{v}_0 - \nabla\mathbf{v}\cdot\mathbf{p}L - \mathbf{v}_G + \dot{\mathbf{p}}L + \mathbf{p}\dot{L})$$
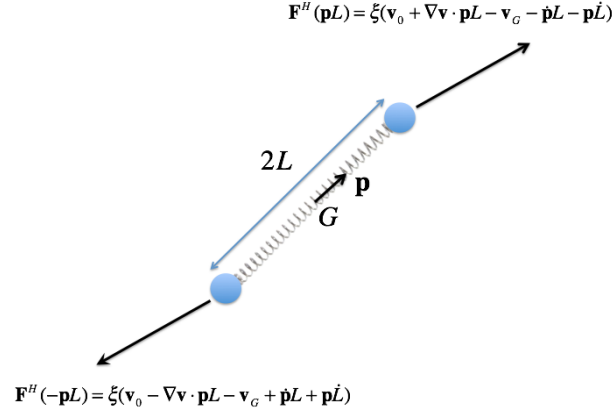
Figure A.1 – Extensible rod immersed in a flow.

expression that introduced into Eq. (A.1) reads

$$\xi(\nabla\mathbf{v}\cdot\mathbf{p}L - \dot{\mathbf{p}}L - \mathbf{p}\dot{L}) = \xi(\nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})L - \dot{L})\mathbf{p}, \tag{A.3}$$

that leads to the rotary velocity $\dot{\mathbf{p}}$

$$\dot{\mathbf{p}} = \nabla\mathbf{v}\cdot\mathbf{p} - \nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})\mathbf{p}, \tag{A.4}$$

that is nothing else than the standard Jeffery expression for ellipsoids of infinite aspect ratio (rods) [Jeffery 1922].

Now, by equating the force acting on the beads λ with the one within the spring, we have

$$2\mathcal{K}(L - L^0) = \xi(\nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})L - \dot{L}), \tag{A.5}$$

or

$$\dot{L} = -\frac{2\mathcal{K}}{\xi}(L - L^0) + \nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})L. \tag{A.6}$$

Thus, the kinematics of an elastic dumbbell of reference length $2L^0$ with conformation at time t given by its orientation $\mathbf{p}$ and length $2L$, read

$$\begin{cases} \dot{\mathbf{p}} = \nabla\mathbf{v}\cdot\mathbf{p} - \nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})\mathbf{p} \\ \dot{L} = -\frac{2\mathcal{K}}{\xi}(L - L^0) + \nabla\mathbf{v}:(\mathbf{p}\otimes\mathbf{p})L \end{cases}. \tag{A.7}$$

A.2.2 From rigid ellipses to orthogonal elastic bi-dumbells

In [Chinesta 2013] it was proven that in order to represent a rigid ellipse whose kinematics are given by the Jeffery equation it suffices to consider a rigid system composed of two mutually orthogonal rods whose lengths correspond with the length of the ellipse axes.

In this case, if $\mathbf{p}$ refers to the direction of the ellipse largest axis, and $\mathcal{F} = \frac{r^2-1}{r^2+1}$, with r the ellipse aspect ratio, we have

$$\dot{\mathbf{p}} = \mathbf{\Omega}\cdot\mathbf{p} + \mathcal{F}(\mathbf{D}\cdot\mathbf{p} - \mathbf{D}:(\mathbf{p}\otimes\mathbf{p})\mathbf{p}), \tag{A.8}$$

where $\mathbf{\Omega}$ and $\mathbf{D}$ are respectively the vorticity and the rate of strain tensors, $2\mathbf{\Omega} = \nabla\mathbf{v} - (\nabla\mathbf{v})^T$ and $2\mathbf{D} = \nabla\mathbf{v} + (\nabla\mathbf{v})^T$.

Now, we address the more general case in which both rigid rods are replaced by two extensible and mutually perpendicular springs of reference lengths L_1^0 and L_2^0. In the sequel, the same reference lengths are assumed for both dumbbells, i.e. $2L_1^0 = 2L_2^0 = 2L^0$.

In the present configuration, and considering that as proven in our former works the centre of gravity moves with the fluid, the hydrodynamic forces applying at beads $L_1\mathbf{p}_1$ and $L_2\mathbf{p}_2$, $\mathbf{F}_1^H$ and $\mathbf{F}_2^H$ read respectively

$$\mathbf{F}_1^H = \xi(\nabla\mathbf{v}\cdot\mathbf{p}_1 L_1 - \dot{\mathbf{p}}_1 L_1 - \mathbf{p}_1\dot{L}_1), \tag{A.9}$$

and

$$\mathbf{F}_2^H = \xi(\nabla\mathbf{v}\cdot\mathbf{p}_2 L_2 - \dot{\mathbf{p}}_2 L_2 - \mathbf{p}_2\dot{L}_2), \tag{A.10}$$

with $\mathbf{p}_1 \perp \mathbf{p}_2$, and with their orientation rates of change expressed from

$$\begin{cases} \dot{\mathbf{p}}_1 = \boldsymbol{\omega}\times\mathbf{p}_1 \\ \dot{\mathbf{p}}_2 = \boldsymbol{\omega}\times\mathbf{p}_2 \end{cases}. \tag{A.11}$$

The angular momentum balance implies now

$$L_1^2\mathbf{p}_1\times(\nabla\mathbf{v}\cdot\mathbf{p}_1 - \dot{\mathbf{p}}_1) + L_2^2\mathbf{p}_2\times(\nabla\mathbf{v}\cdot\mathbf{p}_2 - \dot{\mathbf{p}}_2) = \mathbf{0}, \tag{A.12}$$

which coincides with the expression obtained in the case of rigid rods [Chinesta 2013], proving the validity of the Jeffery equation in the case of orthogonal elastic bi-dumbells.

Introducing the Jeffery equation (A.8) with $r = \frac{L_1}{L_2}$ that implies $\mathcal{F} = \frac{L_1^2 - L_2^2}{L_1^2 + L_2^2}$, i.e.

$$\dot{\mathbf{p}}_1 = \boldsymbol{\Omega}\cdot\mathbf{p}_1 + \mathcal{F}\left(\mathbf{D}\cdot\mathbf{p}_1 - \left(\mathbf{p}_1^T\cdot\mathbf{D}\cdot\mathbf{p}_1\right)\mathbf{p}_1\right), \tag{A.13}$$

into the expression of the hydrodynamic force acting on bead $\mathbf{p}_1 L_1$, we have

$$\mathbf{F}_1^H = \xi(\nabla\mathbf{v}\cdot\mathbf{p}_1 L_1 - \dot{\mathbf{p}}_1 L_1 - \mathbf{p}_1\dot{L}_1) \tag{A.14}$$

$$= \xi\left(\nabla\mathbf{v}\cdot\mathbf{p}_1 L_1 - \boldsymbol{\Omega}\cdot\mathbf{p}_1 L_1 - \mathcal{F}\left(\mathbf{D}\cdot\mathbf{p}_1 L_1 - \left(\mathbf{p}_1^T\cdot\mathbf{D}\cdot\mathbf{p}_1\right)\mathbf{p}_1 L_1\right) - \mathbf{p}_1\dot{L}_1\right) \tag{A.15}$$

$$= \xi L_1\left((1-\mathcal{F})\mathbf{D}\cdot\mathbf{p}_1 + \mathcal{F}\left(\mathbf{p}_1^T\cdot\mathbf{D}\cdot\mathbf{p}_1\right)\mathbf{p}_1\right) - \xi\mathbf{p}_1\dot{L}_1. \tag{A.16}$$

A.2.3 Modelling incompressible ellipses from orthogonal elastic bi-dumbells

The projection of force $\mathbf{F}_1^H$ in the direction $\mathbf{p}_1$ is the one that causes the spring extension, i.e.

$$2\mathcal{K}(L_1 - L^0) - F_1^I = \xi L_1\mathbf{p}_1^T\cdot\mathbf{D}\cdot\mathbf{p}_1 - \xi\dot{L}_1, \tag{A.17}$$

where the force F_1^I ensures the incompressibility constraint. Similar calculations lead to

$$2\mathcal{K}(L_2 - L^0) - F_2^I = \xi L_2\mathbf{p}_2^T\cdot\mathbf{D}\cdot\mathbf{p}_2 - \xi\dot{L}_2, \tag{A.18}$$

where it was assumed that $\mathcal{K}_1 = \mathcal{K}_2 = \mathcal{K}$.

Thus, the extension velocities read

$$\dot{L}_1 = -\frac{2\mathcal{K}}{\xi}(L_1 - L^0) + L_1\mathbf{p}_1^T\cdot\mathbf{D}\cdot\mathbf{p}_1 + \frac{1}{\xi}F_1^I, \tag{A.19}$$

and

$$\dot{L}_2 = -\frac{2\mathcal{K}}{\xi}(L_2 - L^0) + L_2\mathbf{p}_2^T\cdot\mathbf{D}\cdot\mathbf{p}_2 + \frac{1}{\xi}F_2^I. \tag{A.20}$$

Taking into account

$$\begin{cases} \mathbf{p}_1^T \cdot \mathbf{D} \cdot \mathbf{p}_1 = \mathbf{D} : (\mathbf{p}_1 \otimes \mathbf{p}_1) \\ \mathbf{p}_2^T \cdot \mathbf{D} \cdot \mathbf{p}_2 = \mathbf{D} : (\mathbf{p}_2 \otimes \mathbf{p}_2) \end{cases}, \tag{A.21}$$

and the fact that, since $\mathbf{p}_1$ and $\mathbf{p}_2$ are mutually orthogonal, $(\mathbf{p}_1 \otimes \mathbf{p}_1) + (\mathbf{p}_2 \otimes \mathbf{p}_2) = \mathbf{I}$ (with $\mathbf{I}$ the identity tensor), the incompressibility constraint $\frac{d}{dt}(L_1 L_2) = 0$ reads

$$\dot{L}_1 L_2 + L_1 \dot{L}_2 = -\frac{2\mathcal{K}}{\xi}(L_1 - L^0)L_2 - \frac{2\mathcal{K}}{\xi}(L_2 - L^0)L_1 + F_1^I \frac{L_2}{\xi} + F_2^I \frac{L_1}{\xi} = 0, \tag{A.22}$$

which simplifies to

$$\dot{L}_1 L_2 + L_1 \dot{L}_2 = -\frac{4\mathcal{K}}{\xi}L_1 L_2 + \frac{2\mathcal{K}}{\xi}L^0(L_1 + L_2) + F_1^I \frac{L_2}{\xi} + F_2^I \frac{L_1}{\xi} = 0. \tag{A.23}$$

Forces related to the incompressibility constraint are expected to contribute isotropically to the resulting macroscopic stress, and then taken in the flow incompressibility constraint. Thus, if we compute the contribution of F_1^I and F_2^I to the stress by using the Kramers rule, we have

$$\boldsymbol{\sigma}^I = F_1^I \mathbf{p}_1 \otimes L_1 \mathbf{p}_1 + F_2^I \mathbf{p}_2 \otimes L_2 \mathbf{p}_2 = F_1^I L_1 (\mathbf{p}_1 \otimes \mathbf{p}_1) + F_2^I L_2 (\mathbf{p}_2 \otimes \mathbf{p}_2), \tag{A.24}$$

which suggests considering $F_1^I L_1 = F_2^I L_2$, since $\mathbf{p}_1 \otimes \mathbf{p}_1 + \mathbf{p}_2 \otimes \mathbf{p}_2 = \mathbf{I}$.

Thus, considering $F_2^I = F_1^I \frac{L_1}{L_2}$ in Eq. (A.23), we have

$$F_1^I \left(\frac{L_1^2 + L_2^2}{\xi L_2} \right) = \frac{4\mathcal{K}}{\xi}L_1 L_2 - \frac{2\mathcal{K}}{\xi}L^0(L_1 + L_2), \tag{A.25}$$

or

$$F_1^I = 4\mathcal{K}\frac{L_1 L_2^2}{L_1^2 + L_2^2} - 2\mathcal{K}\frac{L^0(L_1 + L_2)L_2}{L_1^2 + L_2^2}, \tag{A.26}$$

that vanishes for the relaxed case $L_1 = L_2 = L^0$, i.e. $F_1^I(L_1 = L_2 = L^0) = 0$ and consequently $F_2^I(L_1 = L_2 = L^0) = 0$.

Thus, finally the governing equations for the orthogonal elastic bi-dumbbell representing an incompressible ellipse read:

$$\begin{cases} \dot{\mathbf{p}}_1 = \boldsymbol{\Omega} \cdot \mathbf{p}_1 + \frac{L_1^2 - L_2^2}{L_1^2 + L_2^2}\mathbf{D} \cdot \mathbf{p}_1 - \frac{L_1^2 - L_2^2}{L_1^2 + L_2^2}\left(\mathbf{p}_1^T \cdot \mathbf{D} \cdot \mathbf{p}_1\right)\mathbf{p}_1 \\ \dot{L}_1 = -\frac{2\mathcal{K}}{\xi}(L_1 - L^0) - \frac{1}{\xi}F_1^I + L_1 \mathbf{p}_1^T \cdot \mathbf{D} \cdot \mathbf{p}_1 \\ \dot{L}_2 = -\frac{L_2}{L_1}\dot{L}_1 \\ F_1^I = 4\mathcal{K}\frac{L_1 L_2^2}{L_1^2 + L_2^2} - 2\mathcal{K}\frac{L^0(L_1 + L_2)L_2}{L_1^2 + L_2^2} \\ F_2^I = F_1^I \frac{L_1}{L_2} \end{cases}. \tag{A.27}$$

A.2.4 Conformation descriptor

When considering control volumes (small enough with respect to the flow but large enough with respect to the cell size) in a flowing foam, it can be observed that the cells in each control volume have similar shapes and orientations. In that case, the cell population in each volume element can be described using $\mathbf{p}_1$ and L_1 (when L^0 is assumed known).

The contribution of a cell to the stress using the Kramers rule can be obtained from the elastic forces. The contribution due to the incompressibility constraint being isotropic, it can be aggregated to the pressure term. Thus, the conformation contribution $\boldsymbol{\sigma}^c$ results

$$\boldsymbol{\sigma}^c = 2\mathcal{K}(L_1 - L^0)\mathbf{p}_1 \otimes L_1 \mathbf{p}_1 + 2\mathcal{K}(L_2 - L^0)\mathbf{p}_2 \otimes L_2 \mathbf{p}_2 \tag{A.28}$$

$$= 2\mathcal{K}(L_1 - L^0)L_1(\mathbf{p}_1 \otimes \mathbf{p}_1) + 2\mathcal{K}(L_2 - L^0)L_2(\mathbf{p}_2 \otimes \mathbf{p}_2) \tag{A.29}$$

$$= 2\mathcal{K}\left(\Delta L_1 L_1(\mathbf{p}_1 \otimes \mathbf{p}_1) + \Delta L_2 L_2(\mathbf{p}_2 \otimes \mathbf{p}_2)\right), \tag{A.30}$$

that vanishes in the relaxed configuration $L_1 = L_2 = L^0$, with $\Delta L_1 = 0$ and $\Delta L_2 = 0$.

In these circumstances, the simplest choice for the conformation tensor consists of the second-order symmetric tensor $\mathbf{c}$ defined from

$$\mathbf{c} = \Delta L_1 L_1 (\mathbf{p}_1 \otimes \mathbf{p}_1) + \Delta L_2 L_2 (\mathbf{p}_2 \otimes \mathbf{p}_2). \tag{A.31}$$

It is important to note that the conformation does not involve an averaging process affecting the different cells inside the control volume because we assumed that all of them share almost the same conformation. However, as it is usual in the modelling of suspensions, this hypothesis could easily be relaxed and we would thus consider the average of different cell conformation, as addressed in the micro-macro simulations by Keunings and coworkers using the Lagrangian Particle Method − LPM − [Halin *et al.* 1998, Keunings 2004].

Even though many rheological behaviours could be associated to flowing foams, this paper focuses on the consideration of induced anisotropy due to the cell deformation. For this reason, and without loss of generality, we consider in what follows the simplest rheological behaviour consisting of a Newtonian behaviour complemented with an elastic contribution related to the cell deformation. In any case, more complex bulk rheologies could be considered, as for example the one related to a viscoplastic behaviour.

Thus, the macroscopic viscoelastic constitutive equation reads

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D} + \mu\mathbf{c}, \tag{A.32}$$

where p is the pressure, that can be viewed as the Lagrange multiplier associated with the macroscopic flow incompressibility constraints, η the effective homogenized fluid viscosity, and μ the rheological parameter affecting the contribution of the microscopic conformation that scales with $\mathcal{K}$ and the volume concentration of cells.

The origin of the elasticity introduced into the conformation field evolution is related to the surface tension that resists the cell deformation from its spherical reference conformation. Topological changes can operate at the elemental cell level. However, on average their net effect is to avoid too large cellular distorsions. Thus, the coefficient μ should describe both the surface tension as well as the microscopic topological changes. In the proposed model, it remains purely phenomenological and should be identified from appropriate rheological tests.

Even though the present model does not address relaxation mechanisms, they could easily be incorporated by including a viscous component at the spring-beads level.

A.2.5 Rheological behaviour

In order to study the rheological response of the proposed model, we carried out a numerical study to obtain the loss and storage moduli of a foam. To proceed, consider a periodic shear strain

$$\gamma(t) = \gamma_0 \sin(\omega t) \tag{A.33}$$

whose time derivative provides the evolution of the shear rate

$$\dot{\gamma}(t) = \gamma_0 \omega \cos(\omega t). \tag{A.34}$$

In simple shear flow, the velocity gradient thus reads $\nabla \mathbf{v} = \begin{bmatrix} 0 & \dot{\gamma}(t) \\ 0 & 0 \end{bmatrix}$, and we could use the model equations (??) to obtain the time evolution of the shape and orientation of a cell subject to such an oscillatory flow. We can thus obtain the time evolution $\mathbf{c}(t)$ of the conformation tensor introduced in the previous section.
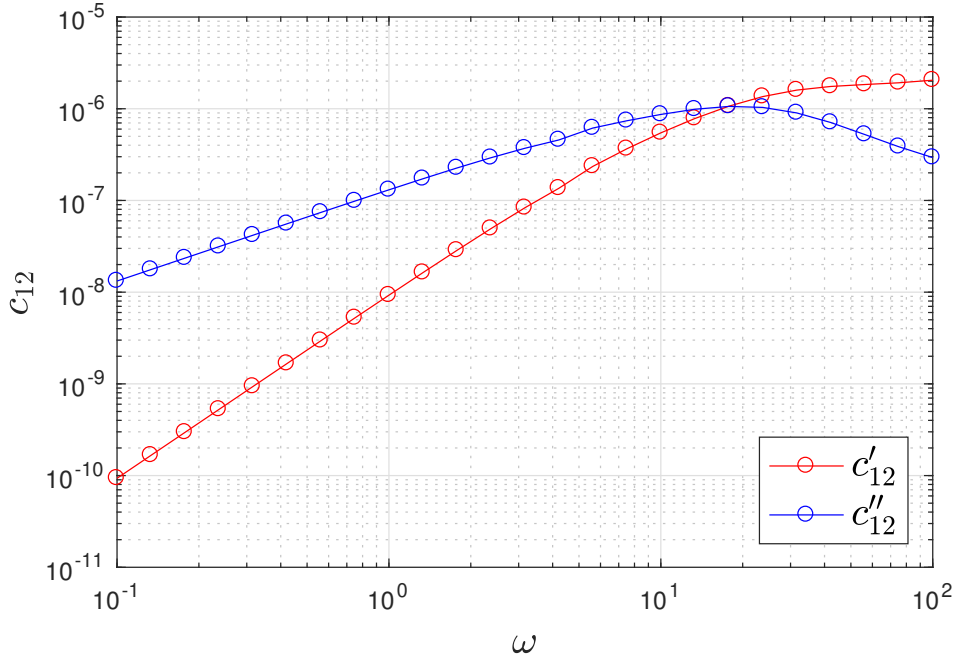
Figure A.2 – Oscillatory shear - c'_{12} and c''_{12} components

We can now decompose the off-diagonal component of tensor $\mathbf{c}(t)$, that is $c_{12}(t)$ in its in-phase and out-of-phase contributions according to

$$c_{12}(t) = c'_{12}\sin(\omega t) + c''_{12}\cos(\omega t), \tag{A.35}$$

where c'_{12} and c''_{12} are analogous to the so-called storage (G') and loss (G'') modulus respectively (they actually correspond to the storage and loss moduli associated with the stress contribution $\boldsymbol{\sigma}^c$).

Figure A.2 shows the value of c'_{12} and c''_{12} for an oscillation angular frequency ranging from 0.1 to 100 rad/s. The parameters are $\xi = 0.1$, $\mathcal{K} = 1$, $L^0 = 0.005$ (metric system of units). We can recognize a Maxwell-like model, with the classical slopes of 1 (G'') and 2 (G') at low frequency.

To obtain properly the storage (G') and loss (G'') moduli, we should also consider the viscous component of the stress tensor. Thus the moduli are given by

$$G' = \frac{\sigma'_{12}}{\gamma_0} = \mu\frac{c'_{12}}{\gamma_0} \tag{A.36}$$

$$G'' = \frac{\sigma''_{12}}{\gamma_0} = 2\eta\omega + \mu\frac{c''_{12}}{\gamma_0}. \tag{A.37}$$

Finally, we also conducted another rheological test to observe the behaviour of the conformation tensor when a constant shear flow is brutally stopped. As expected, we obtain an exponential relaxation of the tensor components towards zero.

A.3 Macroscopic flow problem

Neglecting inertia, the macroscopic flow problem is defined as follows:

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \\ \boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D} + \mu\mathbf{c} \\ \nabla \cdot \mathbf{v} = 0 \end{cases}, \qquad (A.38)$$

where $\boldsymbol{\sigma}$ is the Cauchy's stress tensor, $\mathbf{I}$ the unit tensor and $\mathbf{v}$ the macroscopic fluid velocity field. It can be noticed that the problem reduces to the standard Stokes flow model as soon as $\mu = 0$. Incompressibility is justified by the fact that drainage is neglected.

The flow model must be complemented with the conformation expression

$$\mathbf{c} = \Delta L_1 L_1 (\mathbf{p}_1 \otimes \mathbf{p}_1) + \Delta L_2 L_2 (\mathbf{p}_2 \otimes \mathbf{p}_2), \qquad (A.39)$$

whose evolution is governed by the microstructural model

$$\begin{cases} \dot{\mathbf{p}}_1 = \boldsymbol{\Omega} \cdot \mathbf{p}_1 + \frac{L_1^2 - L_2^2}{L_1^2 + L_2^2} \mathbf{D} \cdot \mathbf{p}_1 - \frac{L_1^2 - L_2^2}{L_1^2 + L_2^2} \left(\mathbf{p}_1^T \cdot \mathbf{D} \cdot \mathbf{p}_1 \right) \mathbf{p}_1 \\ \dot{L}_1 = -\frac{2\mathcal{K}}{\xi}(L_1 - L^0) - \frac{1}{\xi}F^I + L_1 \mathbf{p}_1^T \cdot \mathbf{D} \cdot \mathbf{p}_1 \\ \dot{L}_2 = -\frac{L_2}{L_1}\dot{L}_1 \\ F_1^I = 4\mathcal{K}\frac{L_1 L_2^2}{L_1^2 + L_2^2} - 2\mathcal{K}\frac{L^0(L_1 + L_2)L_2}{L_1^2 + L_2^2} \\ F_2^I = F_1^I \frac{L_1}{L_2} \end{cases}. \qquad (A.40)$$

In order to solve the resulting flow model defined in $\Omega \subset \mathbb{R}^2$, appropriate boundary conditions must be enforced at the domain boundary $\Gamma \equiv \partial\Omega$

$$\begin{cases} \mathbf{v} = \mathbf{v}_g & \text{in } \Gamma_D \\ \mathbf{t} = \boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_g & \text{in } \Gamma_N \end{cases}, \qquad (A.41)$$

with $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

A.4 Numerical solution strategy

Cells are assumed represented at the initial time by a spherical conformation tensor $\mathbf{c} = \mathbf{0}$, since the cells are initially in the relaxed state, $L_1 = L_2 = L^0$.

1. The associated Stokes problem is solved at time t_n, from the microstructural term given at the previous time step $\mu\mathbf{c}(t_{n-1})$. The problem is solved using a standard mixed velocity-pressure formulation using any finite element satisfying the stability conditions, the so-called LBB conditions. In the example reported later, we considered Q9/Q4 finite elements;

2. Then, from the just computed velocity field $\mathbf{v}(t_n)$ the orientation and extension fields, $\mathbf{p}_1(\mathbf{x}, t_n)$ and $L_1(\mathbf{x}, t_n)$ respectively, are updated by using a first-order discontinuous Galerkin formulation;

3. Finally, the conformation tensor is updated $\mathbf{c}(t_n)$.

These three steps are repeated until reaching the maximum simulation time or the steady state.

A.4.1 Updating the conformation field

In the numerical experiments described below, we consider 2D flows defined in 2D geometries $\Omega \subset \mathbb{R}^2$. In the 2D case, the unit vector $\mathbf{p}_1$ can be expressed by $\mathbf{p}_1 = (\cos\theta, \sin\theta)^T$. By

taking the time derivative, we get $\dot{\mathbf{p}}_1 = \dot{\theta}(-\sin\theta, \cos\theta)^T$. Using θ as orientation descriptor, the Jeffery equation can be expressed as

$$\dot{\theta}(-\sin\theta, \cos\theta)^T = \mathbf{G}(\theta, \nabla\mathbf{v}, L_1). \tag{A.42}$$

Multiplying the previous expression by $(-\sin\theta, \cos\theta)$, we obtain the scalar equation

$$\dot{\theta} = \mathcal{G}(\theta, \nabla\mathbf{v}, L_1), \tag{A.43}$$

where the material derivative can be expressed in an Eulerian framework by introducing the orientation field $\theta(\mathbf{x}, t)$ whose evolution is governed by

$$\frac{\partial\theta}{\partial t} + \mathbf{v} \cdot \nabla\theta = \mathcal{G}(\theta, \nabla\mathbf{v}, L_1). \tag{A.44}$$

The scalar equation governing the evolution of the spring length can also be written as

$$\frac{\partial L_1}{\partial t} + \mathbf{v} \cdot \nabla L_1 = \mathcal{H}(\theta, \nabla\mathbf{v}, L_1). \tag{A.45}$$

Both equations (A.44) and (A.45) are purely advective, and thus appropriate discretization taking into account their hyperbolic character must be used. We make here the simplest choice, a first-order discontinuous Galerkin scheme, that considers the generic variable $\mathcal{P}$ (θ or L_1) constant in each element Ω^e belonging to the mesh $\mathcal{M}$ of Ω. The test function is assumed also constant in each element, vanishing outside. The balance in element Ω^e reads, taking into account the flow incompressibility ($\nabla \cdot \mathbf{v} = 0$):

$$\int_{\Omega^e} \frac{\partial\mathcal{P}}{\partial t} d\mathbf{x} + \int_{\Omega^e} \nabla \cdot (\mathbf{v}\mathcal{P}) d\mathbf{x} = \int_{\Omega^e} \mathcal{J} d\mathbf{x}, \tag{A.46}$$

where the source term $\mathcal{J}$ represents $\mathcal{G}$ or $\mathcal{H}$ depending on the considered equation.

Using the divergence theorem, the second term of the left-hand side can be written from the boundary flux, i.e.

$$\int_{\Omega^e} \frac{\partial\mathcal{P}}{\partial t} d\mathbf{x} + \int_{\partial\Omega^e} \mathcal{P}\mathbf{v} \cdot \mathbf{n} \, d\mathbf{x} = \int_{\Omega^e} \mathcal{J} d\mathbf{x}, \tag{A.47}$$

where $\mathbf{n}$ is the unit outward vector normal to the element boundary.

As $\mathcal{P}$ is not defined on $\partial\Omega^e$, we consider the element boundary decomposition $\partial\Omega^e = \partial^+\Omega^e \cup \partial^-\Omega^e$, where $\partial^-\Omega^e$ and $\partial^+\Omega^e$ represents the inflow and outflow element boundaries, both defined from $\mathbf{v} \cdot \mathbf{n} < 0$ and $\mathbf{v} \cdot \mathbf{n} > 0$ respectively. Then we assume that the property on the inflow boundary is given by its value at the upstream element, i.e. $\mathcal{P}(\mathbf{x} \in \partial^-\Omega^e) = \mathcal{P}^{e^-}$, whereas on the outflow element boundary, it is given by the property at element Ω^e, i.e. $\mathcal{P}(\mathbf{x} \in \partial^+\Omega^e) = \mathcal{P}^e$.

A.4.2 Discussion

- In the flow model just proposed there is no size effects, which implies that the characteristic length of cells must be small in relation to the one characterizing the spatial variation of the macroscopic velocity field.

- The mesh considered for integrating the velocity field must be small enough to capture all the macroscopic velocity field details.
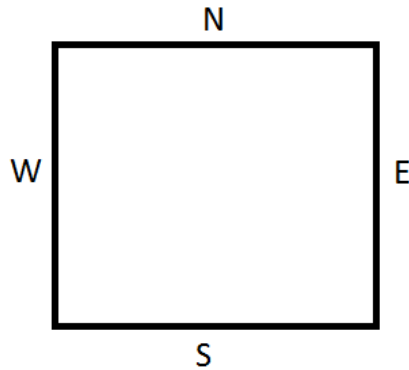
Figure A.3 – 2D domain.

- The mesh considered for calculating the conformation field must be small enough for assuming cells in each element of the mesh described by the conformation field inside the element, and it must be large enough for assuming that it represents a population of cells. However, as soon as the model is described in a continuous way, there is no contradiction with the fact of considering elements smaller that the characteristic size of the cell (in the case of polymer flows one can consider elements smaller that the size of a molecule).

A.5 Numerical results

A.5.1 Uncoupled microstructure-flow calculations

In this section, we address some simple flows in order to evaluate the response of the cells. The model is uncoupled in the sense that the flow induces cell deformation, but the cell conformation does not affect the flow kinematics. Thus, even though some of the flows addressed here could exhibit rich kinematics in practice, in what follows the uncoupled solution does not allow to capture such a rich kinematics, as for example shear banding. Moreover, the only interest of this section being the evolution of the cell deformation and its macroscopic description, rheology is not considered in the analyses carried out.

In the numerical examples considered in this section a unit square is considered as depicted in Fig. A.3. The initial (relaxed) conformation is given by a zero conformation tensor $\mathbf{c}(\mathbf{x}, t = 0) = \mathbf{0}$.

Four different flows are considered: (i) a simple shear; (ii) a contraction flow; (iii) the driven cavity flow problem and finally (iv) the flow around a square obstacle. In all cases, the microstructure was computed on the basis of the associated Stokes kinematics, i.e. considering the uncoupled flow associated with $\mu = 0$.

In order to quantify the way the flow behaviour (shear, rigid motion or elongation) affects the microstructure evolution we consider a flow criterion. For that purpose, first, we introduce the relative rate of rotation $\mathbf{W}$ from

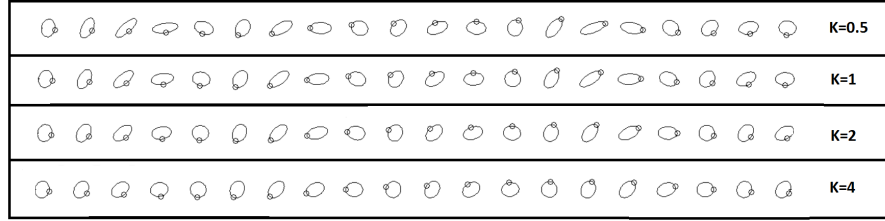$$\mathbf{W} = \boldsymbol{\omega} - \mathbf{w}, \tag{A.48}$$

Figure A.4 – Conformation evolution for different spring stiffness in a simple shear flow.

where $\boldsymbol{\omega}$ is related to the flow vorticity $\boldsymbol{\Omega}$ according to

$$\boldsymbol{\Omega} = \boldsymbol{\varepsilon} \cdot \boldsymbol{\omega}, \tag{A.49}$$

where $\boldsymbol{\varepsilon}$ is the third-order permutation tensor (also known as the Levi-Civita tensor). The vector $\boldsymbol{\omega}$ can also be written in terms of the curl of the velocity,

$$\boldsymbol{\omega} = -\frac{1}{2}\nabla \times \mathbf{v}. \tag{A.50}$$

The vector $\mathbf{w}$ in Eq. (A.48) represents the angular velocity of the eigenvectors of the rate of strain tensor $\mathbf{D}$.

A simple local descriptor of the type of flow can then be constructed from the second invariant of $\mathbf{D}$, $\dot{\gamma} = \sqrt{2\mathbf{D} : \mathbf{D}}$, and from the norm of $\mathbf{W}$, $\|\mathbf{W}\|$, according to

$$\chi = \frac{4\|\mathbf{W}\|}{\dot{\gamma} + 2\|\mathbf{W}\|}. \tag{A.51}$$

We have $0 \leq \chi \leq 2$, and more specifically

$$\chi = \left\{ \begin{array}{ll} 0 & \text{in planar extension,} \\ 1 & \text{in pure shear,} \\ 2 & \text{in rigid motion.} \end{array} \right. \tag{A.52}$$

Figure A.4 shows the evolution of the microstructure conformation for different $\mathcal{K}$ in a pure shear with $\xi = 0.1$ and $L^0 = 0.005$ (metric system of units). The qualitative analysis performed here does not require a precise determination of these coefficients. As it can be noticed, the more stiffness the spring has, the less deformation the conformation presents. In order to prove that the conformation is not only accommodating a deformation but also that it is rotating, we included a small cercle on the ellipsoid surface to appreciate the way in which that point is evolving in time.

A.5.1.1 Simple shear flow

We consider $V = 2\,\mathrm{m}\cdot\mathrm{s}^{-1}$ on the N-boundary, zero velocity on the S-boundary and a linear velocity evolution on the E-boundary and W-boundaries. The microstructure effects were introduced by using $\mathcal{K} = 1$.

Figure A.5 depicts the velocity field, that evolves almost linearly through the domain thickness, as well as the flow criterion, that as expected corresponds to a perfect shear behaviour. The pressure field is constant in the whole flow domain.
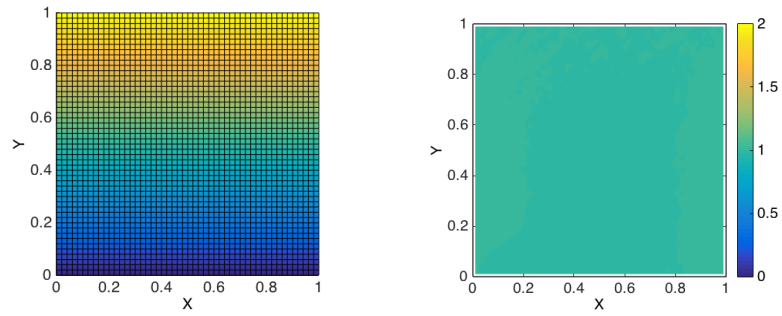
Figure A.5 – Velocity field (left) and flow criterion (right) in a simple shear flow.
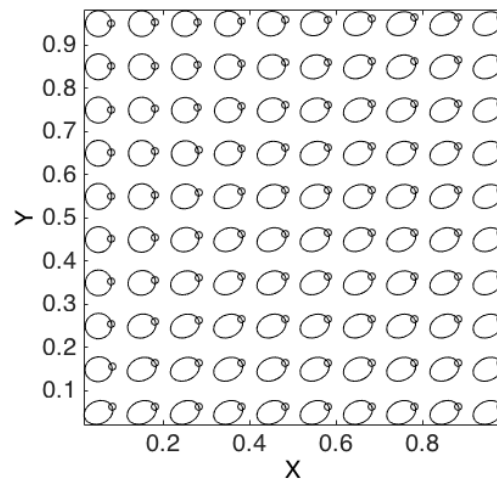
Figure A.6 – Microstructure conformation in a simple shear flow.
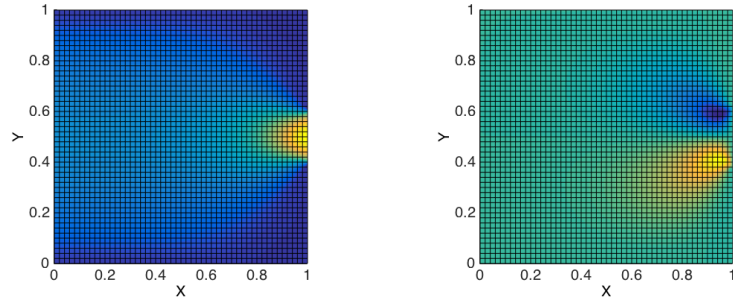
Figure A.7 – Extrusion-like flow problem. Velocity field: (left) x-component and (right) y-component.
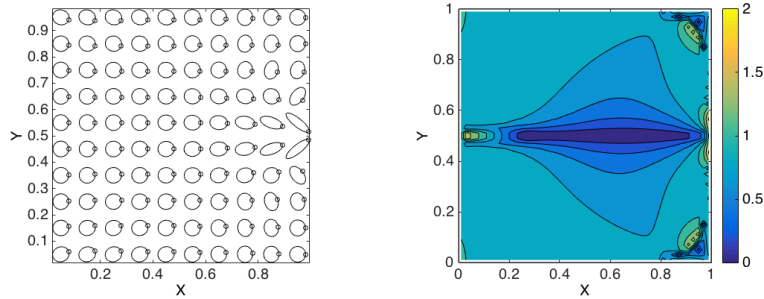
Figure A.8 – Conformation distribution (left) and flow criterion (right) in the extrusion-like flow.

Figure A.6 shows the steady state conformation. Because of the shear, the conformation is expected to rotate clockwise. The microstructure shows a significant variation along the domain as a consequence of the different velocity and constant shear rate.

A.5.1.2 Extrusion-like flow

In the present case, we consider again $\mathcal{K} = 1$. A Poiseuille velocity profile (parabolic) is enforced on the W-boundary $(-2y(y-1), 0)^T$, whereas at the E-boundary the fluid leaves the square domain throughout an exit where tension-free boundary conditions are enforced.

Figure A.7 depicts both components of the velocity field whereas Fig. A.8 shows the steady state conformation and the flow criterion. As it can be seen, the largest conformation axis remains aligned with the streamlines. A pure extension is noticed along the symmetry axis in agreement with the flow criterion.

A.5.1.3 Driven cavity flow

The only difference with respect to the previously analyzed flow is that now a unit horizontal velocity is applied on the top wall (N-boundary) that induces the fluid flow within the cavity.

Figure A.9 depicts both components of the velocity field. Figure A.10 shows the conformation at a given time as well te steady state flow crieria. The conformation evolves periodically and no steady state is reached.
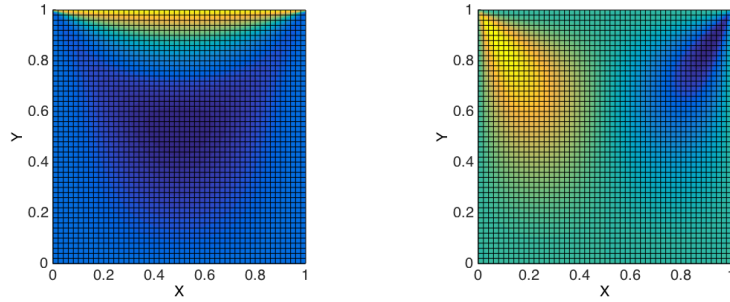
Figure A.9 – Driven cavity flow problem. Velocity field: (left) x-components and (right) y-component.
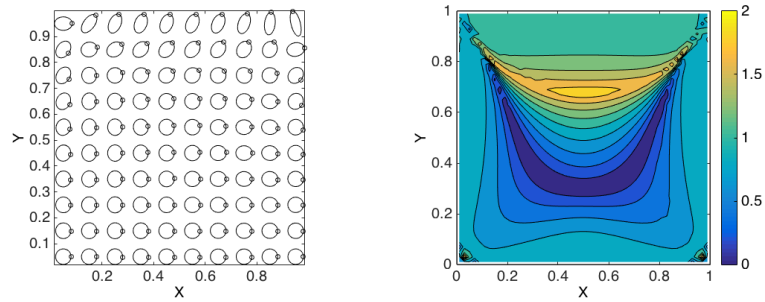
Figure A.10 – Conformation (left) and flow criterion (right) in the driven cavity flow problem.
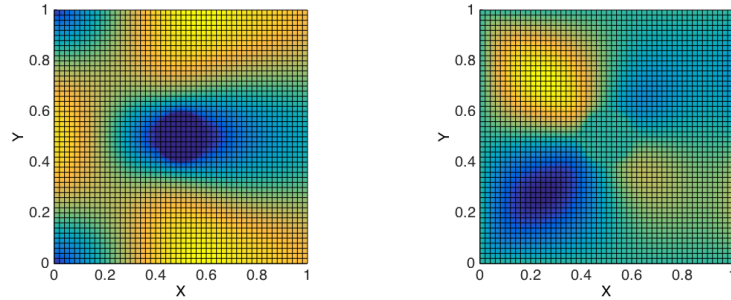
Figure A.11 – Flow around an obstacle. Velocity field: (left) x-component and (right) y-components.
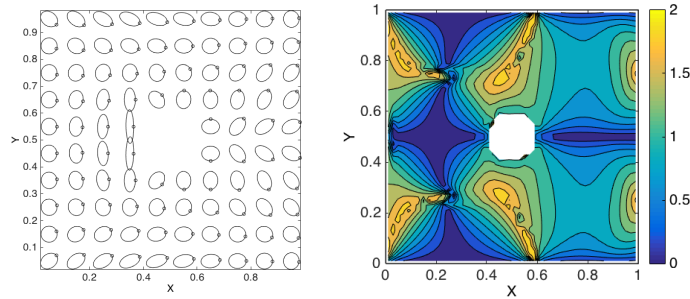
Figure A.12 – Conformation (left) and flow criterion (right) in the flow around an obstacle.

A.5.1.4 Flow around an obstacle

In the present case, a unit horizontal velocity is enforced on the W-boundary, with non-slip conditions on the top and bottom walls and a free boundary condition is specified on the E-boundary. The domain contains in its center an obstacle where non-slip boundary conditions are enforced.

Figure A.11 depicts both components of the velocity field. The velocity field presents two stagnation points located at the intersection between $y = 0.5$ and the obstacle. These stagnation points correspond with the maximum and minimum pressures.

Figure A.12 depicts the conformation and flow criteria. As it can be noticed, elongation is specially present upstream and downstream, the shear being located in the neighbourhood of the upper and lower sides of the obstacle, with the expected effects on the microstructure conformation.

A.5.2 Coupled simulations

This section adresses a coupled simulation in which flow kinematics induces microstructure evolution and the latter affects at its turn the flow kinematics.

Figure A.13 shows two snapshots taken from films of flowing foams where the flow and microstructures were assumed almost at steady-state. The microstructure of these snapshots will be used first to identify the model parameters and then to test the agreement
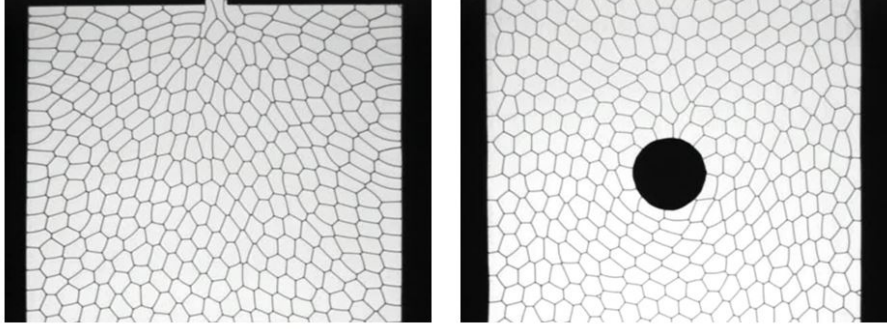
Figure A.13 – Experimental snapshots (courtesy of F. Graner).

between the computed and observed microstructures. Of course, from the information of kinematics only, we cannot conclude on rheological aspects but at least this serves to evaluate the proposed model in terms of conformation evolution.

In order to extract from these images the conformation tensor, both images were segmented (every pixel is transformed into either black or white) and then cell boundaries were easily identified. From that, the centre of gravity of each cell can easily be obtained as well as the conformation tensor. If $\mathbf{X}_i^e$, $i = 1, \ldots, Q^e$ are the points defining the wall of cell $\mathcal{C}^e$, the centre of gravity and the inertia tensor $\mathbf{J}^e$ are obtained from

$$\mathbf{X}_G^e = \frac{1}{Q^e} \sum_{i=1}^{Q^e} \mathbf{X}_i^e \tag{A.53}$$

and

$$\mathbf{J}^e = \frac{1}{Q^e} \sum_{i=1}^{Q^e} (\mathbf{X}_i^e - \mathbf{X}_G^e) \otimes (\mathbf{X}_i^e - \mathbf{X}_G^e) \tag{A.54}$$

respectively.

The numerical inertia tensor can be calculated as soon as $\mathbf{p}_1$ and L_1 are known ($\mathbf{p}_2$ and L_2 derive from $\mathbf{p}_1$ and L_1).

$$\mathbf{J} = L_1^2(\mathbf{p}_1 \otimes \mathbf{p}_1) + L_2^2(\mathbf{p}_2 \otimes \mathbf{p}_2). \tag{A.55}$$

In Fig. A.14, the experimental ellipses related to $\mathbf{J}^e$ are superimposed to the cells, and also to the solution predicted by using the proposed model. The model parameters $\mathcal{K}$, ξ, η and μ are, as previously indicated, chosen in order to obtain cell shapes close to those observed experimentally: (metric system of units) $\mathcal{K} = 1$, $\xi = 0.1$, $\eta = 1$ and $\mu = 0.1$. $L^0 = 0.005$ was identified from the average cell surface observed experimentally. In absence of rheological data, we noticed that many choices of those parameters lead to similar microstructures and that the impact of those on the kinematics was almost negligible. Thus, a proper rheological characterization seems compulsory to obtain an adequate flow-microstructure coupling.

A.6   Conclusions

This work proposes a simple model for flowing foams, where the microstructure is introduced from a conformation tensor that describes the structural elasticity. The numerical
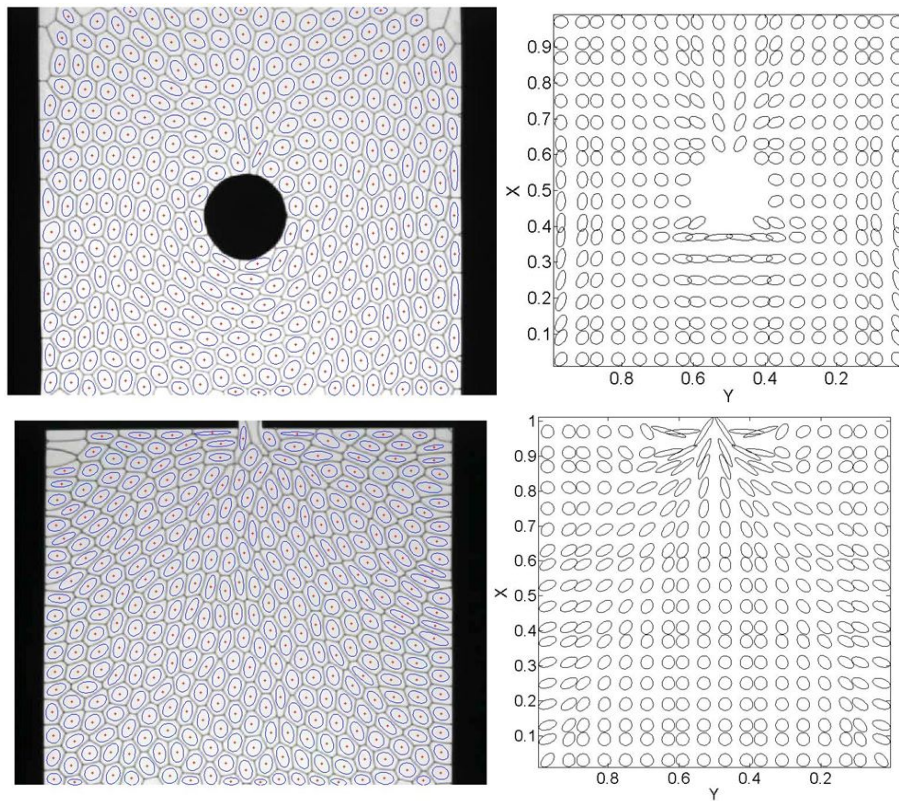
Figure A.14 – Experimental microstructure with the associated conformation super-imposed (left) and the associated numerical predictions (right).

predictions have been compared with some experimental results proving the ability of the model to describe the effective kinematics as well as the flow induced microstructure evolution.

More applicative analyses, in particular in the study of industrial processes, require an appropriate rheological characterization and very probably the proposal of appropriate rheometric devices. All these aspects will be addressed in future works.

Separated representation of the Stokes weak form

The efficient computer implementation of the separated representation constructor discussed in Section 5.3.1.3 needs a separated form of the flow problem expressed in a weak form (5.51). For that purpose we first consider the second term $\mathbf{D}^* : \mathbf{D}$, that takes into account expression (5.49) as follows

$$4\mathbf{D}^* : \mathbf{D} = \nabla\mathbf{v}^* : \nabla\mathbf{v} + \nabla\mathbf{v}^* : (\nabla\mathbf{v})^T + (\nabla\mathbf{v}^*)^T : \nabla\mathbf{v} + (\nabla\mathbf{v}^*)^T : (\nabla\mathbf{v})^T. \tag{B.1}$$

The simplest choice of the test function $\mathbf{v}^*(\mathbf{x}, z)$ is

$$\mathbf{v}^*(\mathbf{x}, z) = \mathbf{P}^*(\mathbf{x}) \circ \mathbf{T}(z) + \mathbf{P}(\mathbf{x}) \circ \mathbf{T}^*(z), \tag{B.2}$$

from which the velocity gradient is:

$$\nabla\mathbf{v}^* = \mathbb{P}^* \circ \mathbb{T} + \mathbb{P} \circ \mathbb{T}^*. \tag{B.3}$$

The choice of $\mathbb{P}$ and $\mathbb{T}$ in Eq. (B.3) is discussed in Section 5.3.1.3.

Developing the first term in Eq. (B.1) (the other terms follow the same rationale) taking into account Eq. (5.47) results

$$\nabla\mathbf{v}^* : \nabla\mathbf{v} \approx (\mathbb{P}^* \circ \mathbb{T} + \mathbb{P} \circ \mathbb{T}^*) : \left(\sum_{j=1}^{N} \mathbb{P}_j \circ \mathbb{T}_j \right). \tag{B.4}$$

It is easy to note that if matrices $\mathbb{M}(\mathbf{x})$ and $\mathbb{N}(\mathbf{x})$ depend on the in-plane coordinates $\mathbf{x}$, and matrices $\mathbb{U}(z)$ and $\mathbb{V}(z)$ depend on the out-of-plane coordinate z, we have

$$(\mathbb{M} \circ \mathbb{U}) : (\mathbb{N} \circ \mathbb{V}) = (\mathbb{M} \circ \mathbb{N}) : (\mathbb{U} \circ \mathbb{V}). \tag{B.5}$$

Using this equality, Eq. (B.4) can be written as

$$\nabla\mathbf{v}^* : \nabla\mathbf{v} \approx \sum_{j=1}^{N} \left\{ (\mathbb{P}^* \circ \mathbb{P}_j) : (\mathbb{T} \circ \mathbb{T}_j) + (\mathbb{P} \circ \mathbb{P}_j) : (\mathbb{T}^* \circ \mathbb{T}_j) \right\}, \tag{B.6}$$

and the other terms involved in Eq. (B.1) as

$$\nabla\mathbf{v}^* : (\nabla\mathbf{v})^T \approx \sum_{j=1}^{N} \left\{ \left(\mathbb{P}^* \circ \mathbb{P}_j^T\right) : \left(\mathbb{T} \circ \mathbb{T}_j^T\right) + \left(\mathbb{P} \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^* \circ \mathbb{T}_j^T\right) \right\}, \tag{B.7}$$

$$(\nabla\mathbf{v})^{*T} : \nabla\mathbf{v} \approx \sum_{j=1}^{N} \left\{ \left(\mathbb{P}^{*T} \circ \mathbb{P}_j\right) : \left(\mathbb{T}^T \circ \mathbb{T}_j\right) + \left(\mathbb{P}^T \circ \mathbb{P}_j\right) : \left(\mathbb{T}^{*T} \circ \mathbb{T}_j\right) \right\}, \tag{B.8}$$

and

$$(\nabla \mathbf{v})^{*T} : (\nabla \mathbf{v})^T \approx \sum_{j=1}^{N} \left\{ \left(\mathbb{P}^{*T} \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^T \circ \mathbb{T}_j^T\right) + \left(\mathbb{P}^T \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^{*T} \circ \mathbb{T}_j^T\right) \right\}. \qquad (B.9)$$

Thus, we finally obtain

$$4\mathbf{D}^* : \mathbf{D} \approx \sum_{j=1}^{N} \left\{ \left(\mathbb{P}^* \circ \mathbb{P}_j\right) : \left(\mathbb{T} \circ \mathbb{T}_j\right) + \left(\mathbb{P} \circ \mathbb{P}_j\right) : \left(\mathbb{T}^* \circ \mathbb{T}_j\right) \right\} +$$

$$\sum_{j=1}^{N} \left\{ \left(\mathbb{P}^* \circ \mathbb{P}_j^T\right) : \left(\mathbb{T} \circ \mathbb{T}_j^T\right) + \left(\mathbb{P} \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^* \circ \mathbb{T}_j^T\right) \right\} +$$

$$\sum_{j=1}^{N} \left\{ \left(\mathbb{P}^{*T} \circ \mathbb{P}_j\right) : \left(\mathbb{T}^T \circ \mathbb{T}_j\right) + \left(\mathbb{P}^T \circ \mathbb{P}_j\right) : \left(\mathbb{T}^{*T} \circ \mathbb{T}_j\right) \right\} +$$

$$\sum_{j=1}^{N} \left\{ \left(\mathbb{P}^{*T} \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^T \circ \mathbb{T}_j^T\right) + \left(\mathbb{P}^T \circ \mathbb{P}_j^T\right) : \left(\mathbb{T}^{*T} \circ \mathbb{T}_j^T\right) \right\} =$$

$$\sum_{j=1}^{N} \sum_{k=1}^{4} \left\{ \mathbb{A}_{jk}^*(\mathbf{x}) : \mathbb{B}_{jk}(z) + \mathbb{A}_{jk}(\mathbf{x}) : \mathbb{B}_{jk}^*(z) \right\}, \qquad (B.10)$$

where $\forall j,\, j = 1, \cdots, N$

$$\mathbb{A}_{jk}^* = \begin{cases} \mathbb{P}^* \circ \mathbb{P}_j, & \text{if } k = 1 \\ \mathbb{P}^* \circ \mathbb{P}_j^T, & \text{if } k = 2 \\ \mathbb{P}^{*T} \circ \mathbb{P}_j, & \text{if } k = 3 \\ \mathbb{P}^{*T} \circ \mathbb{P}_j^T, & \text{if } k = 4 \end{cases}, \qquad (B.11)$$

$$\mathbb{B}_{jk} = \begin{cases} \mathbb{T} \circ \mathbb{T}_j, & \text{if } k = 1 \\ \mathbb{T} \circ \mathbb{T}_j^T, & \text{if } k = 2 \\ \mathbb{T}^T \circ \mathbb{T}_j, & \text{if } k = 3 \\ \mathbb{T}^T \circ \mathbb{T}_j^T, & \text{if } k = 4 \end{cases}, \qquad (B.12)$$

$$\mathbb{A}_{jk} = \begin{cases} \mathbb{P} \circ \mathbb{P}_j, & \text{if } k = 1 \\ \mathbb{P} \circ \mathbb{P}_j^T, & \text{if } k = 2 \\ \mathbb{P}^T \circ \mathbb{P}_j, & \text{if } k = 3 \\ \mathbb{P}^T \circ \mathbb{P}_j^T, & \text{if } k = 4 \end{cases}, \qquad (B.13)$$

and

$$\mathbb{B}_{jk}^* = \begin{cases} \mathbb{T}^* \circ \mathbb{T}_j, & \text{if } k = 1 \\ \mathbb{T}^* \circ \mathbb{T}_j^T, & \text{if } k = 2 \\ \mathbb{T}^{*T} \circ \mathbb{T}_j, & \text{if } k = 3 \\ \mathbb{T}^{*T} \circ \mathbb{T}_j^T, & \text{if } k = 4 \end{cases}, \qquad (B.14)$$

On the other hand, the first term in Eq. (5.51) can be expressed as:

$$\text{Tr}(\mathbf{D}^*) \cdot \text{Tr}(\mathbf{D}) = \text{Tr}(\nabla \mathbf{v}^*) \cdot \text{Tr}(\nabla \mathbf{v}), \qquad (B.15)$$

with

$$\text{Tr}(\nabla \mathbf{v}) \approx \left(\sum_{i=1}^{N} \mathbb{P}_i \circ \mathbb{T}_i \right) : \mathbf{I}. \qquad (B.16)$$

Thus, a generic term in Eq. (B.15) can be written as

$$((\mathbb{P}^* \circ \mathbb{T} + \mathbb{P} \circ \mathbb{T}^*) : \mathbf{I}) \cdot ((\mathbb{P}_j \circ \mathbb{T}_j) : \mathbf{I}) . \tag{B.17}$$

Now, by defining $\mathcal{V}(\mathbb{J})$ the vector form of the diagonal matrix $\mathbb{J}$, Eq. (B.17) results

$$((\mathbb{P}^* \circ \mathbb{T} + \mathbb{P} \circ \mathbb{T}^*) : \mathbf{I}) \cdot ((\mathbb{P}_j \circ \mathbb{T}_j) : \mathbf{I}) =$$

$$(\mathcal{V}(\mathbb{P}^* \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{P}_j \circ \mathbf{I})) : (\mathcal{V}(\mathbb{T} \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{T}_j \circ \mathbf{I})) +$$

$$(\mathcal{V}(\mathbb{P} \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{P}_j \circ \mathbf{I})) : (\mathcal{V}(\mathbb{T}^* \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{T}_j \circ \mathbf{I})), \tag{B.18}$$

that allows finally casting the first term in the weak form (5.51) as

$$\mathrm{Tr}(\mathbf{D}^*) \cdot \mathrm{Tr}(\mathbf{D}) \approx \sum_{j=1}^{N} \mathbb{F}_j^*(\mathbf{x}) : \mathbb{G}_j(z) + \mathbb{F}_j(\mathbf{x}) : \mathbb{G}_j^*(z) \tag{B.19}$$

with

$$\mathbb{F}_j^* = \mathcal{V}(\mathbb{P}^* \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{P}_j \circ \mathbf{I}), \tag{B.20}$$

$$\mathbb{G}_j = \mathcal{V}(\mathbb{T} \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{T}_j \circ \mathbf{I}), \tag{B.21}$$

$$\mathbb{F}_j = \mathcal{V}(\mathbb{P} \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{P}_j \circ \mathbf{I}), \tag{B.22}$$

and

$$\mathbb{G}_j^* = \mathcal{V}(\mathbb{T}^* \circ \mathbf{I}) \otimes \mathcal{V}(\mathbb{T}_j \circ \mathbf{I}). \tag{B.23}$$

Bibliography

[Ammar *et al.* 2006] A. Ammar, B. Mokdad, F. Chinesta and R. Keunings. *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids.* J. Non-Newtonian Fluid Mech., **139**, 153–176, 2006.

[Ammar *et al.* 2007] A. Ammar, B. Mokdad, F. Chinesta and R. Keunings. *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representations.* J. Non-Newtonian Fluid Mech., **144**, 98–121, 2007.

[Ammar *et al.* 2011] Amine Ammar, Francisco Chinesta, Elias Cueto and Manuel Doblare. *Proper Generalized Decomposition of Time-Multiscale Models.* International Journal for Numerical Methods in Engineering, 2011.

[Ammar *et al.* 2016] A. Ammar, E. Abisset-Chavanne, F. Chinesta and R. Keunings. *Flow modelling of quasi-Newtonian fluids in two-scale fibrous fabrics: Advanced simulations.* International Journal of Material Forming, 2016.

[Amsallem & Farhat 2008] D. Amsallem and Ch. Farhat. *An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity.* AIAA Journal, **46**, 1803–1813, 2008.

[Babuska & Melenk 1996] I. Babuska and J. M. Melenk. *The partition of Unity Finite Element Method: Basic Theory and Applications.* Comp. Meth. in Appl. Mech. and Eng., **4**, 289–314, 1996.

[Babuska & Melenk 1997] I. Babuska and J. M. Melenk. *The partition of Unity Method.* International Journal for Numerical Methods in Engineering, **40**, 727–758, 1997.

[Badías *et al.* 2017] Alberto Badías, David González, Iciar Alfaro, Francisco Chinesta and Elias Cueto. *Local proper generalized decomposition.* International Journal for Numerical Methods in Engineering, **112**, 1715–1732, 2017. nme.5578.

[Benito *et al.* 2008] S. Benito, C.-H. Bruneau, T. Colin, C. Gay and F. Molino. *An elasto-visco-plastic model for immortal foams or emulsions.* European Physics Journal E, 225–251, 2008.

[Bensaid *et al.* 2005] S. Bensaid, D. Trichet and J. Fouladgar. *3D simulation of induction heating of anisotropic composite materials/ IEEE Transactions of Magnetics.* IEEE Transactions of Magnetics, **41**, 1568–1571, 2005.

[Bensaid *et al.* 2006] S. Bensaid, D. Trichet and J. Fouladgar. *Electromagnetic and thermal behaviours of multi-layer anisotropic composite materials.* IEEE Transactions of Magnetics, **42**, 995–998, 2006.

[Bhushan 2001] B. Bhushan. Modern tribology handbook. CRC Press, 2001.

[Bikard *et al.* 2005] J. Bikard, J. Bruchon, T. Coupez and B. Vergnes. *Numerical prediction of the foam structure of polymeric materials by direct 3D simulation of their expansion by chemical reaction based on a multidomain method.* Journal of Materials Science, **40**, 5875–5881, 2005.

[Binetruy *et al.* 2015] C. Binetruy, F. Chinesta and R. Keunings. *Flows in Polymers, Reinforced Polymers and Composites. A multiscale approach.* Springerbriefs, Springer, 2015.

[Bognet *et al.* 2012] 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, **201-204**, $1 - 12$, 2012.

[Bognet *et al.* 2014] B. Bognet, A. Leygue and F. Chinesta. *Separated representations of 3D elastic solutions in shell geometries.* Advanced Modelling and Simulation in Engineering Sciences, **1**, 2014.

[Bohm 2009] H. Bohm. *A Short Introduction to Basic Aspects of Continuum Micromechanics. ISLB Report 208.* Technical report, TU Wien, Vienna, 2009.

[Borst 2008] R. Borst. *Challenges in computational materials science: Multiple scales, multi-physics and evolving discontinuities.* Computational Materials Science, 43/1, 1-15, 2008.

[Borzacchiello *et al.* 2017a] Domenico Borzacchiello, José V. Aguado and Francisco Chinesta. *Non-intrusive Sparse Subspace Learning for Parametrized Problems.* Archives of Computational Methods in Engineering, 2017.

[Borzacchiello *et al.* 2017b] Domenico Borzacchiello, Jose Vicente Aguado and Francisco Chinesta. *Reduced order modelling for efficient numerical optimisation of a hot-wall chemical vapour deposition reactor.* International Journal of Numerical Methods for Heat & Fluid Flow, **27**, 1602–1622, 2017.

[Boyse *et al.* 1992] W.E. Boyse, D.R. Lynch, K.D. Paulsen and G.N. Minerbo. *Nodal-based finite-element modeling of Maxwell equations.* IEEE Transactions on Antennas and Propagation, **40**, 642–651, 1992.

[Breitkopf *et al.* 2005] Piotr Breitkopf, Hakim Naceur, Alain Rassineux and Pierre Villon. *Moving least squares response surface approximation: Formulation and metal forming applications.* Computers & Structures, **83**, 1411–1428, 2005.

[Brunton *et al.* 2016] Steven L. Brunton, Joshua L. Proctor and J. Nathan Kutz. *Discovering governing equations from data by sparse identification of nonlinear dynamical systems.* Proceedings of the National Academy of Sciences, 2016.

[Bui *et al.* 2016] H.K. Bui, G. Wasselynck, D. Trichet and G. Berthiau. *Degenerated hexahedral whitney elements for electromagnetic fields computation in multi-layer anisotropic thin regions.* IEEE Transactions of Magnetics, **52**, 2016.

[Bungartz & Griebel 2004] Hans-Joachim Bungartz and Michael Griebel. *Sparse grids.* Acta numerica, **13**, 147–269, 2004.

[Bur *et al.* 2016] N. Bur, P. Joyot, Ch. Ghnatios, P. Villon, E. Cueto and F. Chinesta. *Advanced computational vademecums for Automated Fibre Placement processes.* Advanced Modelling and Simulation in Engineering Sciences, 2016.

[Cheddadi *et al.* 2008] I. Cheddadi, P. Saramito, C. Raufaste, P. Marmottant and F. Graner. *Numerical modelling of foam Couette flows.* European Physical Journal E, **27**, 123–133, 2008.

[Chew & Nasir 1989] W.C. Chew and M.A. Nasir. *A Variational analysis of anisotropic, inhomogeneous dielectric waveguides,.* IEEE Transactions of Microwaves Theory and Technologies, **37**, 661–668, 1989.

[Chinesta & Cueto 2014] Francisco Chinesta and Elias Cueto. Pgd-based modeling of materials, structures and processes. Springer International Publishing Switzerland, 2014.

[Chinesta & Ladeveze 2014] F. Chinesta and P. Ladeveze, editors. Separated Representations and PGD-Based Model Reduction. Springer International Publishing, 2014.

[Chinesta *et al.* 2008] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene and F. Boust. *Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization.* Computer Methods in Applied Mechanics and Engineering, **197**, 400–413, 2008.

[Chinesta *et al.* 2010] F. Chinesta, A. Ammar and E. Cueto. *Recent advances in the use of the Proper Generalized Decomposition for solving multidimensional models.* Archives of Computational Methods in Engineering, **17**, 327–350, 2010.

[Chinesta *et al.* 2011a] 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, 395–404, 2011.

[Chinesta *et al.* 2011b] Francisco Chinesta, Pierre Ladeveze and Elias Cueto. *A Short Review on Model Order Reduction Based on Proper Generalized Decomposition.* Archives of Computational Methods in Engineering, **18**, 395–404, 2011.

[Chinesta *et al.* 2013a] F. Chinesta, A. Leygue, F. Bordeu, J. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar and A. Huerta. *Parametric PGD based computational vademecum for efficient design, optimization and control.* Archives of Computational Methods in Engineering, 20/1, 31-59, 2013.

[Chinesta *et al.* 2013b] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar and A. Huerta. *PGD-Based Computational Vademecum for Efficient Design, Optimization and Control.* Archives of Computational Methods in Engineering, **20**, 31–59, 2013.

[Chinesta *et al.* 2014] Francisco Chinesta, Roland Keunings and Adrien Leygue. The proper generalized decomposition for advanced numerical simulations. Springer International Publishing Switzerland, 2014.

[Chinesta *et al.* 2017] F. Chinesta, A. Huerta, G. Rozza and K. Willcox. *Model Reduction Methods.* In Encyclopedia of Computational Mechanics. Wiley Online Library, second édition, 2017.

[Chinesta *et al.* 2018] Francisco Chinesta, Elias Cueto, Emmanuelle Abisset-Chavanne, Jean-Louis Duval and Fouad El Khaldi. *Virtual, Digital and Hybrid Twins: A New Paradigm in Data-Based Engineering and Engineered Data.* Archives of Computational Methods in Engineering, 2018.

[Chinesta 2013] F. Chinesta. *From single-scale to two-scales kinetic theory descriptions of rods suspensions.* Archives in Computational Methods in Engineering, **20**, 1–29, 2013.

[Crespo *et al.* 2017] José Crespo, Marcos Latorre and Francisco Javier Montáns. *WYPI-WYG hyperelasticity for isotropic, compressible materials.* Computational Mechanics, **59**, 73–92, 2017.

[Cueto *et al.* 2016] E. Cueto, D. González and I. Alfaro. Proper generalized decompositions: An introduction to computer implementation with matlab. SpringerBriefs in Applied Sciences and Technology. Springer International Publishing, 2016.

[Dagnall 2014] H. Dagnall. *Exploring Surface Texture.* Taylor Hobson Pusblishing Ltd., 2014.

[Darema 2005] F. Darema. *Grid Computing and Beyond: The Context of Dynamic Data Driven Applications Systems.* Proceedings of the IEEE, **93**, 692–697, 2005.

[David & Ladeveze 2010] Neron David and Pierre Ladeveze. *Proper generalized decomposition for multiscale and multiphysics problems.* Archives of Computational Methods in Engineering, 2010.

[Epstein *et al.* 2011] Charles Epstein, Gunnar Carlsson and Herbert Edelsbrunner. *Topological data analysis.* Inverse Problems, **27**, 120201, 2011.

[Ericksen 1959] J.L. Ericksen. *Anisotropic Fluids.* Archive for Rational Mechanics and Analysis, 231–237, 1959.

[Everson & Sirovich 1995] R. Everson and L. Sirovich. *Karhunen–Loève procedure for gappy data.* J. Opt. Soc. Am. A, **12**, 1657–1664, 1995.

[E.X. Liu 2016] E.P. Li. E.X. Liu X.C. Wei. *2.5D methodologies for electronic package and PCB modeling: Review and latest development.* IEEE International Symposium on Electromagnetic Compatibility (EMC), 2016.

[Feyel 1999] F. Feyel. *Multiscale FE2 elastoviscoplastic analysis of composite structures.* Computational Materials Science, 16/1-4, 344-354, 1999.

[Feyel 2003] F. Feyel. *A multilevel finite element method (FE2) to describe the response of highly non-linear structures using generalized continua.* Computer Methods in Applied Mechanics and Engineering, **192**, 3233–3244, 2003.

[Fish & Yuan. 2005] J. Fish and Z. Yuan. *Multiscale enrichment based on partition of unity.* International Journal for Numerical Methods in Engineering, 62/10, 1341-1359, 2005.

[Fish 006] J. Fish. *Bridging the scales in nano engineering and science.* Journal of Nanoparticle Research, 8/5, 577-594, 2006.

[Fritzen & Leuschner 2013] Felix Fritzen and Matthias Leuschner. *Reduced basis hybrid computational homogenization based on a mixed incremental formulation.* Computer Methods in Applied Mechanics and Engineering, 2013.

[Geers *et al.* 2010] M. Geers, V. Kouznetsova and W. Brekelmans. *Multi-scale computational homogenization: Trends and challenges.* Journal of Computational and Applied Mathematics, 234/7, 2175-2182, 2010.

[Ghnatios *et al.* 2016] Ch. Ghnatios, E. Abisset-Chavanne, Ch. Binetruy, F. Chinesta and S. Advani. *3D modeling of squeeze flow of multiaxial laminates.* Journal of Non-Newtonian Fluid Mechanics, **234**, 188–200, 2016.

[Giner *et al.* 2013] E. Giner, B. Bognet, J.J. Rodenas, A. Leygue, J. Fuenmayor and F. Chinesta. *The Proper Generalized Decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics.* International Journal of Solid Structures, **50**, 1710–1720, 2013.

[Gonzalez *et al.* 2010] D. Gonzalez, A. Ammar, F. Chinesta and E. Cueto. *Recent advances on the use of separated representations.* International Journal for Numerical Methods in Engineering, **81**, 2010.

[Gonzalez *et al.* 2014] David Gonzalez, Elias Cueto and Francisco Chinesta. *Real-time direct integration of reduced solid dynamics equations.* Internatinal Journal for Numerical Methods in Engineering, **99**, 633–653, 2014.

[González *et al.* 2015] David González, Elías Cueto and Francisco Chinesta. *Computational Patient Avatars for Surgery Planning.* Annals of Biomedical Engineering, **44**, 35–45, 2015.

[González *et al.* 2016] D. González, J. V. Aguado, E. Cueto, E. Abisset-Chavanne and F. Chinesta. *kPCA-Based Parametric Solutions Within the PGD Framework.* Archives of Computational Methods in Engineering, 1–18, 2016.

[Gonzalez *et al.* 2017] David Gonzalez, Alberto Badias, Iciar Alfaro, Francisco Chinesta and Elias Cueto. *Model order reduction for real-time data assimilation through Extended Kalman Filters.* Computer Methods in Applied Mechanics and Engineering, **326**, $679 - 693$, 2017.

[Gonzalez *et al.* 2018a] David Gonzalez, Francisco Chinesta and Elias Cueto. *Learning corrections for hyperelastic models from data.* Frontiers Materials, **, in review**, 2018.

[González *et al.* 2018b] David González, Francisco Chinesta and Elías Cueto. *Thermodynamically consistent data-driven computational mechanics.* Continuum Mechanics and Thermodynamics, 2018.

[Gonzalez *et al.* 2019] David Gonzalez, Francisco Chinesta and Elias Cueto. *Thermodynamically consistent data-driven computational mechanics.* Continuum Mechanics and Thermodynamics, 2019.

[Gyselinck *et al.* 2016] J. Gyselinck, P. Dular, L. Krahenbuhl and R.V. Sabariego. *Finite-element homogenization of laminated iron cores with inclusion of net circulating currents due to imperfect insulation.* IEEE Transactions on Magnetics, **52**, 2016.

[Halabi *et al.* 2013] F. El Halabi, D. González, A. Chico and M. Doblaré. *{FE2} multiscale in linear elasticity based on parametrized microscale models using proper generalized decomposition.* Computer Methods in Applied Mechanics and Engineering, **257**, $183 - 202$, 2013.

[Halin *et al.* 1998] P. Halin, G. Lielens, R. Keunings and V. Legat. *The Lagrangian particle method for macroscopic and micro-macro viscoelastic flow computations,.* Journal of Non-Newtonian Fluid Mechanics, **79**, 387–403, 1998.

[Hazard & Lenoir 996] C. Hazard and M. Lenoir. *On the solution of the time-harmonic scattering problems for Maxwell equations.* SIAM Journal on Mathematical Analysis, **27**, 1597–1630, 1996.

[Hughes & Liu 1978] T.J.R. Hughes and W.K. Liu. *Implicit-explicit finite elements in transient analysis: Stability theory.* Journal of Applied Mechanics, **45**, 371–374, 1978.

[Hughes *et al.* 1998] T. J. R. Hughes, G. R. Feijóo, L. Mazzei and J-B. Quincy. *The variational multiscale method - a paradigm for computational mechanics.* Comput. Methods Appl. Mech. Eng., **166**, 3–24, 1998.

[I. Temizer 2007a] P. Wriggers I. Temizer. *An adaptive method for homogenization in orthotropic nonlinear elasticity.* Computer Methods in Applied Mechanics and Engineering, 196/35-36, 3409-3423, 2007.

[I. Temizer 2007b] T. Zohdi I. Temizer. *A numerical method for homogenization in nonlinear elasticity.* Computational Mechanics. 40/2, 281-298, 2007.

[Ibañez *et al.* 2018] R. Ibañez, E. Abisset-Chavanne, D. Gonzalez, J.L. Duval, E. Cueto and F. Chinesta. *Hybrid constitutive modeling: Data-driven learning of corrections to plasticity models.* International Journal of Material Forming, **accepted for publication**, 2018.

[J.C. Michel 2004] P. Suquet J.C. Michel. *Computational analysis of nonlinear composite structures using the nonuniform transformation field analysis.* Computer Methods in Applied Mechanics and Engineering, 193/48-51, 5477-5502, 2004.

[Jeffery 1922] G.B. Jeffery. *The motion of ellipsoidal particles immersed in a viscous fluid.* Proceedings of the Royal Society London, **A102**, 161–179, 1922.

[Jin 2002] J. Jin. The finite element method in electromagnetics. John Wiley and Sons, 2002.

[Kaiser *et al.* 2018] E. Kaiser, J. N. Kutz and S. L. Brunton. *Sparse identification of nonlinear dynamics for model predictive control in the low-data limit.* Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, **474**, 2018.

[Kanoute *et al.* 2009] P. Kanoute, D. Boso, J. Chaboche and B. Schrefler. *Multiscale Methods for Composites: A Review.* Archives of Computational Methods in Engineering, 16/1, 31-75, 2009.

[Karhunen 1946] K. Karhunen. *Uber lineare methoden in der wahrscheinlichkeitsrechnung.* Ann. Acad. Sci. Fennicae, ser. Al. Math. Phys., **37**, 1946.

[Karimi *et al.* 2017] M. Karimi, H. Droghetti and D.L. Marchisio. *PUFoam: A novel open-source CFD solver for the simulation of polyurethane foams.* Computer Physics Communications, **217**, 138–148, 2017.

[Keunings 2004] R. Keunings. *Micro-macro methods for the multiscale simulation viscoelastic flow using molecular models of kinetic theory.* In: Binding DM, Walters K (eds) Rheology reviews. British Society of Rheology, 67–98, 2004.

[Kirchdoerfer & Ortiz 2016] T. Kirchdoerfer and M. Ortiz. *Data-driven computational mechanics.* Computer Methods in Applied Mechanics and Engineering, **304**, 81 − 101, 2016.

[Kirchdoerfer & Ortiz 2017] T. Kirchdoerfer and M. Ortiz. *Data-driven computing in dynamics*. International Journal for Numerical Methods in Engineering, **113**, 1697–1710, 2017.

[Kutz 2013] J. N. Kutz. Data-driven modeling & scientific computation. methods for complex systems & big-data. Oxford University Press, 2013.

[Ladeveze 1985] P. Ladeveze. *On a family of algorithms for structural mechanics (in french)*. Comptes Rendus Académie des Sciences Paris, **300/2**, 41–44, 1985.

[Ladeveze 1989] P. Ladeveze. *The large time increment method for the analyze of structures with nonlinear constitutive relation described by internal variables*. Comptes Rendus Académie des Sciences Paris, **309**, 1095–1099, 1989.

[Ladeveze 1999] P. Ladeveze. Nonlinear computational structural mechanics. Springer, N.Y., 1999.

[Lamari *et al.* 2010] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin and F. Chinesta. *Routes for Efficient Computational Homogenization of Non-Linear Materials Using the Proper Generalized Decomposition*. Archives of Computational Methods in Engineering, **17/4**, 373–391, 2010.

[Lee & Verleysen 2007] J.A. Lee and M. Verleysen. Nonlinear dimensionality reduction. Springer Verlag, 2007.

[Leon *et al.* 2018] A. Leon, A. Barasinski, E. Abisset-Chavanne, E. Cueto and F. Chinesta. *Wavelet-based multiscale proper generalized decomposition*. Comptes Rendus Academie de Sciences – Mécanique, **346**, 485–500, 2018.

[Liu *et al.* 2016] Zeliang Liu, M.A. Bessa and Wing Kam Liu. *Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials*. Computer Methods in Applied Mechanics and Engineering, **306**, 319 – 341, 2016.

[Loève 1963] M. M. Loève. Probability theory. The University Series in Higher Mathematics, 3rd ed. Van Nostrand, Princeton, NJ, 1963.

[Lopez *et al.* 2015] E. Lopez, E. Abisset-Chavanne, S. Comas-Cardona, C. Binetruy and F. Chinesta. *Flow modeling of linear and nonlinear fluids in two and three scale fibrous fabrics.* . International Journal of Material Forming., 2015.

[Lopez *et al.* 2016a] E. Lopez, D. Gonzalez, J. V. Aguado, E. Abisset-Chavanne, E. Cueto, C. Binetruy and F. Chinesta. *A Manifold Learning Approach for Integrated Computational Materials Engineering*. Archives of Computational Methods in Engineering, 1–10, 2016.

[Lopez *et al.* 2016b] E. Lopez, D. Gonzalez, J.V. Aguado, E. Abisset-Chavanne, F. Lebel, R. Upadhyay, E. Cueto, C. Binetruy and F. Chinesta. *Real-time, image-based, simulation of multiscale models based on locally linear embedding*. Archives of Computational Methods in Engineering, **submitted**, 2016.

[Lopez *et al.* 2016c] E. Lopez, A. Leygue, E. Abisset-Chavanne, S. Comas-Cardona, C. Aufrere, C. Binetruy and F. Chinesta. *Flow modeling of linear and nonlinear fluids in two scale fibrous fabrics. Advanced simulations*. International Journal of Material Forming, 2016.

[Maaten & Hinton 2008] Laurens van der Maaten and Geoffrey Hinton. *Visualizing data using t-SNE*. Journal of machine learning research, **9**, 2579–2605, 2008.

Bibliography

[Maday & Turinici 2002] Yvon Maday and Gabriel Turinici. *A parareal in time procedure for the control of partial differential equations.* Comptes Rendus Mathematique, 2002.

[Mangan *et al.* 2016] N. M. Mangan, S. L. Brunton, J. L. Proctor and J. N. Kutz. *Inferring Biological Networks by Sparse Identification of Nonlinear Dynamics.* IEEE Transactions on Molecular, Biological and Multi-Scale Communications, **2**, 52–63, 2016.

[Mann & Kutz 2016] Jordan Mann and J. Nathan Kutz. *Dynamic mode decomposition for financial trading strategies.* Quantitative Finance, **16**, 1643–1655, 2016.

[McVeigh & Liu 2008] C. McVeigh and W. Liu. *Linking microstructure and properties through a predictive multiresolution continuum.* Computer Methods in Applied Mechanics and Engineering, 197/41-42, 3268-3290, 2008.

[Mena *et al.* 2015] Andres Mena, David Bel, Iciar Alfaro, David Gonzalez, Elias Cueto and Francisco Chinesta. *Towards a pancreatic surgery simulator based on model order reduction.* Advanced Modeling and Simulation in Engineering Sciences, **2**, 31, 2015.

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

[Meyer & Matthies 2003] M. Meyer and H. G. Matthies. *Efficient model reduction in nonlinear dynamics using the Karhunen-Loève expansion and dual-weighted-residual methods.* Computational Mechanics, **31**, 179–191, 2003.

[Michel *et al.* 1999] J. Michel, H. Moulinec and P. Suquet. *Effective properties of composite materials with periodic microstructure: a computational approach.* Computer methods in applied mechanics and engineering, 1999.

[Millán & Arroyo 2013] Daniel Millán and Marino Arroyo. *Nonlinear manifold learning for model reduction in finite elastodynamics.* Computer Methods in Applied Mechanics and Engineering, **261-262**, 118 − 131, 2013.

[Mur 1994] G. Mur. *Edge elements, their advantages and their disadvantages.* IEEE Transactions on Magnetics, **30**, 3552–3557, 1994.

[Mur 1998] G. Mur. *The fallacy of edge elements.* IEEE Transactions on Magnetics, **34**, 3244–3247, 1998.

[Nedelec 1980] J.C. Nedelec. *Mixed finite elements in R3.* Numerische Mathematik, **35**, 315–341, 1980.

[Niroomandi *et al.* 2010] Siamak Niroomandi, Iciar Alfaro, Elias Cueto and Francisco Chinesta. *Model order reduction for hyperelastic materials.* International Journal for Numerical Methods in Engineering, **81**, 1180–1206, 2010.

[Niyonzima *et al.* 2013] I. Niyonzima, R.V. Sabariego, P. Dular, F. Henrotte and C. Geuzaine. *Computational homogenization for laminated ferromagnetic cores in magnetodynamics.* IEEE Transactions on Magnetics, **49**, 2049–2052, 2013.

[Nouy 010] Anthony Nouy. *A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations.* COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING, **199**, 1603–1626, 2010.

[Otin 2010] R. Otin. *Regularized Maxwell equations and nodal finite elements for electromagnetic field computations.* Journal Electromagnetics, **30**, 190–204, 2010.

[Otin 2013] R. Otin. *ERMES: A nodal-based finite element code for electromagnetic simulations in frequency domain.* Computer Physics Communications, **184**, 2588–2595, 2013.

[P. Benner 2016] K. Willcox P. Benner S. Gugercin. *A survey of projection-based model reduction methods for parametric dynamical systems.* SIAM Review, **in press**, 2016.

[Pan & Tan 1997] G. Pan and J. Tan. *General edge element approach to lossy and dispersive structures in anisotropic media.* IEEE Proceedings Microwave Antennas Propagation, **144**, 81–90, 1997.

[Park & Cho 1996] H. M. Park and D. H. Cho. *The use of the Karhunen-Loève decomposition for the modeling of distributed parameter systems.* Chemical Engineering Science, **51**, 81–98, 1996.

[Patera & Rozza 2007] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* Technical report, MIT Pappalardo Monographs in Mechanical Engineering, 2007.

[Paulsen *et al.* 1987] K.D. Paulsen, D.R. Lynch and J.W. Strohbehn. *Numerical treatment of boundary conditions at points connecting more than two electrically distinct regions.* Communications in Applied Numerical Methods, **3**, 53–62, 1987.

[Paulsen *et al.* 1988] K.D. Paulsen, D.R. Lynch and J.W. Strohbehn. *Three dimensional finite, boundary, and hybrid element solutions of Maxwell equations for lossy dielectric media.* IEEE Transactions on Microwave Theory and Techniques, **36**, 682–693, 1988.

[Peherstorfer & Willcox 2015] Benjamin Peherstorfer and Karen Willcox. *Dynamic data-driven reduced-order models.* Computer Methods in Applied Mechanics and Engineering, **291**, 21 − 41, 2015.

[Peherstorfer & Willcox 2016] Benjamin Peherstorfer and Karen Willcox. *Data-driven operator inference for nonintrusive projection-based model reduction.* Computer Methods in Applied Mechanics and Engineering, **306**, 196 − 215, 2016.

[Perez *et al.* 2016] M. Perez, A. Scheuer, E. Abisset-Chavanne, F. Chinesta and R. Keunings. *A multi-scale description of orientation in confined suspensions involving rods.* Journal of Non-Newtonian Fluid Mechanics, 233, 61-74, 2016.

[Polito & Perona 2001] Marzia Polito and Pietro Perona. *Grouping and Dimensionality Reduction By Locally Linear Embedding.* In Advances in Neural Information Processing Systems 14, 1255–1262. MIT Press, 2001.

[Pruliere *et al.* 010] E. Pruliere, F. Chinesta and A. Ammar. *On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition.* Mathematics and Computers in Simulation, **81**, 791–810, 2010.

[qnd D. Dureisseix & Cartraud 2009] A. Mobasher-Amini qnd D. Dureisseix and P. Cartraud. *Multi-scale domain decomposition method for large-scale structural analysis with a zooming technique: Application to plate assembly.* International Journal for Numerical Methods in Engineering, 79/4, 417-443, 2009.

[Quarteroni *et al.* 2011] A. Quarteroni, G. Rozza and A. Manzoni. *Certified reduced basis approximation for parametrized PDE and applications.* J. Math in Industry, **3**, 2011.

[Roweis & Saul 2000] Sam T. Roweis and Lawrence K. Saul. *Nonlinear Dimensionality Reduction by Locally Linear Embedding.* Science, **290**, 2323–2326, 2000.

[Rozza *et al.* 2008] G. Rozza, D.B.P. Huynh and A.T. Patera. *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations – application to transport and continuum mechanics.* Archives of Computational Methods in Engineering, **15/3**, 229–275, 2008.

[Rozza 2014] G. Rozza. *Fundamentals of reduced basis method for problems governed by parametrized PDEs and applications.* In P. Ladeveze and F. Chinesta, editors, CISM Lectures notes "Separated Representation and PGD based model reduction: fundamentals and applications". Springer Verlag, 2014.

[Ryckelynck *et al.* 2006] D. Ryckelynck, F. Chinesta, E. Cueto and A. Ammar. *On the a priori Model Reduction: Overview and recent developments.* Archives of Computational Methods in Engineering, **12(1)**, 91–128, 2006.

[Ryssel & Brunn 1999] E. Ryssel and P. Brunn. *Comparison of a quasi-Newtonian fluid with a viscoelastic fluid in planar contraction flow..* Journal of Non-Newtonian Fluid Mechanichs, 86, 309-335, 1999.

[S. Cohen-Addad 2013] Olivier Pitois. S. Cohen-Addad R. Hohler. *Flow in foams and flowing foams.* Annual Review Fluid Mechanics, **45**, 241–267, 2013.

[S. Fernandez 2004] A. Huerta S. Fernandez. *Imposing essential boundary conditions in mesh-free methods.* Computer Methods in Applied Mechanics and Engineering, **193**, 1257–1275, 2004.

[Scheuer *et al.* 2016] A. Scheuer, E. Abisset-Chavanne, F. Chinesta and R. Keunings. *Second-gradient modelling of orientation development and rheology of confined suspensions.* Journal of Non-Newtonian Fluid Mechanics, 237, 54-64, 2016.

[Scholkopf *et al.* 1999] B. Scholkopf, A. Smola and K.R. Muller. *Kernel principal component analysis.* In ADVANCES IN KERNEL METHODS - SUPPORT VECTOR LEARNING, 327–352. MIT Press, 1999.

[Spencer 2000] A.J.M. Spencer. *Theory of fabric-reinforced viscous fluids.* Composites Part A, **31**, 1311–1321, 2000.

[Strouboulis *et al.* 2004] T. Strouboulis, L. Zhang and I. Babuska. *p-version of the generalized FEM using mesh-based handbooks with applications to multiscale problems.* International Journal for Numerical Methods in Engineering, 60/10, 1639 - 1672, 2004.

[Sun *et al.* 2000] J.S. Sun, K.H. Lee and H.P. Lee. *Comparison of implicit and explicit finite element methods for dynamic problems.* Journal of Materials Processing Technology, **105**, 110–118, 2000.

[Swanson & Hoefer 2003] D.G. Swanson and W.J.R. Hoefer. *Microwave circuit modeling using electromagnetic field simulation.* Artech House, 2003.

[T. Zohdi 2005] P. Wriggers T. Zohdi. An introduction to computational micromechanics. Springer, 2005.

[Tenenbaum *et al.* 2000] J. B. Tenenbaum, V. de Silva and J. C. Langford. *A global framework for nonlinear dimensionality reduction.* Science, **290**, 2319–2323, 2000.

[Thompson & Mendes 2005] R. Thompson and P. Souza Mendes. *Persistence of straining and flow classification.* International Journal of Engineering Science, 43, 79-105, 2005.

[Tibshirani 1996] Robert Tibshirani. *Regression Shrinkage and Selection via the Lasso.* Journal of the Royal Statistical Society. Series B (Methodological), **58**, 267–288, 1996.

[Tlili *et al.* 2015] S. Tlili, C. Gay, F. Graner, P. Marcq, F. Molino and P. Saramito. *Mechanical formalisms for tissue dynamics.* European Physical Journal E, **38**, 33–63, 2015.

[Torquato 2002] S. Torquato. *Statistical Description of Microstructures.* Annual Review of Materials Research, 2002.

[Volkwein 2001] S. Volkwein. *Model reduction using proper orthogonal decomposition.* Technical report, Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz, 2001.

[Wang 2012] Quan Wang. *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models.* CoRR, **abs/1207.3538**, 2012.

[Wasselynck *et al.* 2010] G. Wasselynck, D. Trichet, B. Ramdane and J. Fould. *Interaction between electromagnetic field and CFRP materials: A new multiscale homogenization approach.* IEEE Transactions of Magnetics, **46**, 3277–3280, 2010.

[Yvonnet *et al.* 2009] J. Yvonnet, D. Gonzalez and Q.-C. He. *Numerically explicit potentials for the homogenization of nonlinear elastic heterogeneous materials.* Computer Methods in Applied Mechanics and Engineering, 198/33-36, 2723-2737, 2009.