Réduction de modèle dans une intégration locale-globale, application aux points de soudure pour la simulation des crash-tests.

Agathe Reille

HESAM UNIVERSITÉ

Arts et Métiers Sciences et Technologies

ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE L'INGÉNIEUR
[PIMM – Campus de Paris]

THÈSE

présentée par : **Agathe REILLE**

soutenue le : **13 décembre 2021**

pour obtenir le grade de : **Docteur d'HESAM Université**

préparée à : **École Nationale Supérieure d'Arts et Métiers**

Spécialité : Mécanique, génie mécanique, génie civil

Reduced-order modeling in a local-global integration, application to spot weld for crash-test simulation

THÈSE dirigée par :
[Professeur CHINESTA Francisco]

et co-encadrée par :
[Docteur DAIM Fatima]

Jury

M. Fodil MERAGHNI, Professeur aux Arts et Métiers	Président
M. Carl LABERGERE, Professeur à l'Université de Technologie de Troyes	Rapporteur
M. Antonio FALCO, Professeur à l'Université CEU Cardenal Herrera	Rapporteur
Mme Marianne BERINGHIER, Maître de conférences à l'ISAE ENSMA	Examinatrice
Mme. Christine ROYER, Docteur, Expert chez Renault	Examinatrice
M. Mustapha ZIANE, Docteur, Expert chez ESI Group	Examinateur
Mme. Fatima DAIM, Docteur, Expert chez ESI Group	Examinatrice
M. Francisco CHINESTA, Professeur aux Arts et Métiers	Examinateur
M. Yves TOURBIER, Docteur, Expert chez Renault	Invité

T H È S E

Abstract

Performing a crash test simulation quickly and accurately remains a challenge for car manufacturers despite the generalization of high-performance computing and the progress in numerical methods. Indeed, engineers introduce mesoscale models of small components in the car model to obtain excellent prediction accuracy, but the integration of these local models increases computational time. Model order reduction methods that exploit the generally low dimensionality of the loading space of a physical phenomenon and the correlations in the loading are interesting to improve the accuracy of a crash test model at a low computational cost. In this thesis, we address the reduction of mesoscale spot weld models in ESI Group's high-fidelity crash test model to decrease the computational time of a simulation. Professor Chinesta and Doctor Daim supervised this work conducted inside the consortium dedicated to model order reduction for industrial problems, comprising Gestamp, Renault, ESI Group, and Arts et Métiers.

We propose (i) a new data-driven reduction technique called incremental Dynamic Mode Decomposition (iDMD) and (ii) an integration scheme, the local-global approach, to introduce this reduced model into a finite element model. We applied these techniques to reduce a mesoscale model of a spot weld. The use case represents an assembly of two metal sheets subject to tensile and shear stresses until failure. In less than 10 minutes, we performed a simulation comparable to the reference simulation performed in 1h30, with the same computing power. The accuracy of the reduced model is excellent as long as the damage is not advanced. During the rupture, the prediction error remains low even with a reduced training compared to the problem's dimensionality. Additionally, we integrated the spot weld reduced model into a global model represent-

ing two sheets using the local-global approach. We obtained results comparable to a high-fidelity simulation integrating a spot weld mesoscale model. Thus, the iDMD and the local-global approach associated with a crash-test model would integrate the spot welds' local behaviors without explicitly describing them. Following these encouraging results, we started a more extensive study with Renault on national supercomputers (Jean Zay). These supercomputers allow us to explore the parametric loading space encountered during a crash test to integrate our approach in the dedicated ESI Group software.

This thesis is divided into four chapters. In the first chapter, we introduce the crash test simulation and present our state of the art on model order reduction. In the second chapter, we present the iDMD and the local-global approach. In the third chapter, we present the use case on spot welds. Finally, the last chapter is the conclusion of our approach.

Résumé

Réaliser rapidement et précisément une simulation de crash-test reste un défi pour les constructeurs automobiles malgré la généralisation du calcul haute-performance et les progrès des méthodes numériques. En effet, des modèles à méso-échelle de petits composants sont intégrés dans le modèle global de la voiture pour obtenir une excellente précision sur l'ensemble mais l'introduction de ces modèles locaux augmente fortement le temps d'exécution du calcul. Les méthodes de réduction de modèle qui exploitent la dimension généralement faible de l'ensemble des sollicitations d'un phénomène physique et les corrélations entre celles-ci sont intéressantes pour améliorer la précision d'un modèle de crash-test à moindre coût. Cette thèse traite de la réduction des modèles méso-échelle des points de soudure dans le modèle de crash-test haute-fidélité d'ESI Group afin de diminuer le temps de calcul d'une simulation. Ce travail est supervisé par le professeur Chinesta et le docteur Daim dans le cadre du consortium dédié à la réduction de modèle pour des problèmes industriels comprenant Gestamp, Renault, ESI Group et les Arts et Métiers.

Nous proposons (i) une nouvelle technique de réduction à partir de données appelée incremental Dynamic Mode Decomposition (iDMD) et (ii) un schéma d'intégration, l'approche locale-globale, introduisant ce modèle réduit dans un modèle éléments finis. Nous avons utilisé ces méthodes pour réduire un modèle méso-échelle de point de soudure. Le cas d'application représente un assemblage de deux tôles soumises à des sollicitations mécaniques de traction et de cisaillement jusqu'à rupture. Nous avons réalisé en moins de 10 minutes une simulation comparable à la simulation de référence réalisée en 1h30, à puissance de calcul égale. La précision du modèle réduit est excellente tant que le dommage n'est pas avancé. Lors de la rupture, l'erreur de prédiction

reste faible même avec un entrainement modeste comparé à la dimensionnalité du problème. Par ailleurs, nous avons réalisé l'intégration du modèle réduit d'un point de soudure dans un modèle global représentant deux tôles suivant l'approche locale-globale et obtenu des résultats comparables à une simulation haute-fidélité intégrant un modèle méso-échelle de point de soudure. Ainsi, l'iDMD et l'approche locale-globale associées dans un modèle de crash-test permettraient d'intégrer les comportements locaux des points de soudure sans les décrire explicitement. À la suite de ces résultats encourageants, nous avons débuté une étude plus approfondie avec Renault sur des supercalculateurs nationaux (Jean Zay). Ceux-ci permettent d'explorer l'espace paramétrique des chargements rencontrés lors d'un crash-test afin d'intégrer notre approche dans le logiciel dédié d'ESI Group.

Cette thèse se décompose en quatre chapitres. Le premier chapitre introduit la simulation de crash-test et présente notre état de l'art sur la réduction de modèle. Le deuxième chapitre présente l'iDMD et l'approche locale-globale. Le troisième chapitre expose le cas d'usage sur un point de soudure. Finalement, le quatrième chapitre présente les conclusions sur notre approche.

Résumé étendu

Chapitre 1 : introduction

Les ingénieurs utilisent quotidiennement les crashs virtuels et les essais de chocs internes pour vérifier que la conception du véhicule en cours valide les exigences de sécurité. La simulation des essais d'impacts est l'outil principal de validation car celui-ci permet de réduire les coûts et les délais de développement. Actuellement, le résultat d'un calcul lancé le soir est obtenu le lendemain car une simulation dure une dizaine d'heures (sur une centaine de cœurs). Les constructeurs souhaiteraient améliorer la précision des calculs en maintenant le temps de calcul à moins d'une journée. Leur objectif ultime est de n'accomplir que des essais réels de confirmation de la conception avant les crashs tests d'homologation autorisant la mise sur le marché. Cependant, si l'on améliore la précision du modèle du crash test le temps de calcul augmente démesurément. En effet, pour réaliser des simulations haute-fidélités de crash tests, des modèles locaux de petites zones fragiles doivent être introduits dans le modèle global de la voiture mais ceux-ci allongent la durée du calcul de la simulation.

Dans l'exposé du contexte industriel de cette thèse, nous introduisons le solveur développé par ESI Group pour calculer des simulations de crash test. Celui-ci suit un schéma numérique qui approche la solution d'un problème mathématiques représentant la physique d'un crash test. Le schéma numérique employé pour simuler la dynamique rapide des essais d'impact induit des calculs longs lorsque des modèles méso-échelle sont intégrés pour réaliser des simulations haute-fidélités. Dans cette thèse, nous nous sommes concentrés sur l'intégration des comportements précis des points de soudure, des éléments d'assemblages des tôles très utilisés dans l'industrie automobile.

Ils représentent 80 % des assemblages de la structure d'une voiture. Pour accélérer le calcul de crash lorsque des modèles précis des points de soudure sont introduits dans le modèle du crash, nous avons exploré les techniques de réduction de modèle et de sub-modeling. La réduction de modèle est utilisée pour diminuer le temps de calcul d'un modèle précis et les techniques de sub-modeling servent à introduire des modèles locaux simplifiés dans des modèles éléments finis.

La première section de ce chapitre détaille le rôle des crashs test virtuels dans le développement des projets automobiles. La deuxième section décrit les méthodes numériques du logiciel de crash d'ESI Group. La troisième section se concentre sur les modèles numériques des points de soudre pour le crash. Enfin, la dernière section aborde les techniques de réduction de modèle et de sub-modeling.

Des essais d'impacts à la simulation

La sécurité des passagers et la consommation de carburant sont des objectifs centraux de l'industrie automobile. En effet, une voiture doit répondre à des exigences de consommation et de sécurité pour entrer sur le marché. Pour limiter la consommation des voitures, les ingénieurs allègent les véhicules. Cette stratégie est très intéressante car elle s'applique aussi bien aux moteurs thermiques qu'aux moteurs électriques. Pour assurer la sécurité des passagers, les ingénieurs développent des équipements de protection et renforcent la structure de la voiture, ce qui augmente inévitablement son poids. Face au durcissement des restrictions de consommation, les constructeurs automobiles ont besoin d'outils d'optimisation de la conception très performants pour parvenir à un compromis optimal sur la masse.

La simulation de crash test est l'outil principal pour répondre à ce besoin. En effet, les ingénieurs évaluent quotidiennement la conception en réalisant des essais réels et virtuels. Cependant, les essais réels sont plus contraignants en termes de coûts et de délais que la simulation. Par exemple, un essai de choc en phase amont de conception peut coûter plusieurs centaines de milliers d'euros alors qu'une simulation s'élève seulement à quelques milliers d'euros. De plus, la fabrication d'un prototype pour l'essai de choc peut prendre plusieurs mois, ainsi, lors de l'essai, la conception évaluée n'est plus celle en cours. Pour limiter les essais internes, les constructeurs souhaitent

développer le calcul rapide des simulations haute-fidélité.

Les méthodes numériques dans le logiciel de ESI

ESI Group développe un outil efficace pour le prototypage virtuel appelé Virtual Performance Solution (VPS). Il contient un module pour la simulation de crash test. Ces simulations sont le résultat de calculs effectués selon un schéma numérique sur des unité de calcul haute performance (en anglais High-Performance Computing (HPC)). Ce schéma numérique donne une solution à un problème mathématiques représentant la physique d'un choc. Selon la précision souhaitée, un calcul peut durer des heures, des jours, voire des semaines. Cependant, dans un projet industriel de développement de voitures, les calculs des essais de chocs haute-fidélité ne devraient pas durer plus d'une journée pour s'intégrer dans le calendrier du projet.

L'équation de la dynamique des milieux continus (éq. 1.1) est utilisée pour le calcul de crash tests dans VPS. La solution du problème est approchée avec la méthode des éléments finis selon un schéma temporel explicite utilisé pour la dynamique rapide (algo. 1). Ce schéma numérique est stable si la condition de Courant (éq. 1.4) est respectée. Ainsi une discrétisation localement fine diminue la taille du pas de temps Δt et par conséquent, augmente le temps de calcul.

Les modèles éléments finis précis des points de soudure dans le calcul de crash utilisent une discrétisation spatiale minuscule comparée à celle de la structure de la voiture. Une voiture mesure en moyenne 2,5m de long alors qu'un point de soudure mesure en général 3mm de diamètre. Ainsi, l'introduction de ces modèles est problématique vis-à-vis du temps de calcul accordé pour la simulation lors de la conception d'un véhicule.

Modélisation des points de soudure

Le soudage par points est l'un des principaux procédés d'assemblage dans une voiture. Une voiture contient 4000 à 6000 points de soudure. Les ingénieurs utilisent des modèles numériques simplifiés pour réduire le temps de calcul d'une simulation d'impact. Par exemple, une simulation de crash test utilisant des points de soudure simplifiés s'exécute en 10 heures sur un ordinateur à 140 cœurs en 2020 [Gst+20]. Cependant, ces modèles

simplifient le comportement inélastique des points de soudure avec les effets associés sur l'ensemble de la structure [Don+05]. Par exemple, ils ne prédisent pas l'initiation de fissures dont la propagation dans les tôles fragilise la structure lors d'un choc. En effet, ces modèles ne prennent pas en compte les phénomènes thermiques et mécaniques du procédé de soudage par point qui impactent l'élasticité, la plasticité, le dommage et la rupture de l'assemblage.

Dans l'industrie automobile, des robots soudeurs assemblent les tôles sur les lignes de production. Ces robots viennent localement compresser les tôles à l'aide d'une pince munie d'électrodes. Un fort courant traverse alors celles-ci et le métal fond localement par effet Joule ce qui créé un point de soudure. Les phénomènes physiques (thermiques, mécaniques et métallurgiques) mis en jeu pour achever un point de soudure impactent fortement son comportement dynamique. Trois zones ayant des propriétés mécaniques différentes sont généralement distinguées dans un point de soudure. Le métal de base dans lequel la température ne s'est pas suffisamment élevée pour modifier la microstructure (la structure de la matière à l'échelle microscopique). La zone affectée thermiquement dans laquelle l'élévation de la température est à l'origine de transformations de la structure de la matière sans atteindre la fusion. Finalement, la zone fondue dans laquelle un noyau de métal liquide se forme pendant la chauffe.

Des modèles haute-résolution de points de soudure ont été proposés pour améliorer la précision des prédictions des comportements dynamiques. Ces modèles utilisent une discrétisation fine et 3D du point de soudure pour représenter sa richesse thermomécanique. Ils représentent finement le métal de base, la zone affectée thermiquement et la zone fondue, c'est pourquoi leur intégration dans les solveurs entraîne un effort de calcul déraisonnable. Nous nous sommes donc intéressés à la réduction de modèle pour intégrer les comportements précis prédits par les modèles méso-échelle dans la simulation de crash.

La réduction de modèle

Le modèle éléments finis régissant la dynamique de la structure de la voiture considère tout chargement dans un espace vectoriel de dimension N_{dof}, avec N_{dof} le nombre de degrés de liberté du modèle éléments finis. En pratique, le système n'est pas soumis à

toutes les possibilités de chargement. Par conséquent, la solution $\mathbf{U}(t^n) \in \mathbb{R}^{N_{dof}}$ qui est l'approximation discrète des déplacements $\mathbf{u}(t^n)$ à l'instant t^n se trouve sur une variété dont la dimension est inférieure au nombre de degrés de liberté du système. Les techniques de réduction de modèle exploitent ces observations et reposent sur l'hypothèse que la solution $\mathbf{U}(t^n)$ peut-être définie sur un sous-espace de dimension n, avec $n \ll \mathtt{N}_{dof}$.

Les techniques de réduction construisent des modèles réduits fournissant une représentation en basse dimension d'un modèle en haute dimension, tels que les modèles par éléments finis, avec une précision comparable mais plus rapidement. Deux grandes familles de méthode de réduction existent, les méthodes "a priori" et les méthodes "a posteriori". Contrairement aux premières, les méthodes "a posteriori" construisent le modèle réduit à partir d'instantanés, communément appelés "snapshots". Elles peuvent donc plus facilement être introduites dans un code existant, c'est pourquoi nous nous sommes concentrés dans notre état de l'art sur ces méthodes. Notre approche développée dans le chapitre 2 s'inspire de la "Dynamic Mode Decomposition" (DMD). Cette méthode construit une application linéaire entre deux séquences d'instantanés de solutions décalés dans le temps.

Pour introduire le modèle réduit dans le modèle élément fini nous nous sommes intéressés aux méthodes de sub-moddeling, en particulier à la condensation statique. Cette méthode décompose le domaine d'intérêt en deux domaines : un local finement discrétisé et un global grossièrement maillé. Ils sont séparés par une frontière (un ensemble de nœuds). Le comportement du domaine local est réexprimé sur la frontière pour retirer la discrétisation fine du modèle et augmenter la taille du pas de temps.

En nous inspirant de ces méthodes, nous avons développé deux méthodes numériques pour approcher la précision des effets du modèle fin 3D du point de soudure sur le domaine global de la voiture au pas de temps du domaine global.

Chapitre 2 : apprentissage et couplage du modèle réduit

Nous avons proposé une nouvelle technique de réduction, l'"incremental Dynamic Mode Decomposition" (iDMD) et une méthode de couplage pour introduire le modèle

réduit dans le solveur de crash test. Ce chapitre présente la construction du modèle réduit et son introduction dans un solveur éléments finis.

Nous avons proposé deux techniques de réduction dans le cadre de l'iDMD. Ces travaux sont publiés dans [Rei+19]. Ces deux approches sont fondées sur les données. La première construit un modèle réduit global, tandis que la second construit un modèle réduit local. Cette deuxième technique est adaptée pour modéliser les comportements fortement non linéaires des points de soudure lors des essais de chocs. Dans ce résumé nous concentrons sur cette deuxième approche qui est appliquée sur un cas test dans le chapitre 3. En revisitant la condensation statique, nous proposons également un couplage du modèle réduit avec le solveur.

Dans la première section de ce chapitre, nous introduisons d'abord la procédure d'apprentissage du modèle réduit global, le modèle rang-n. Dans la deuxième section, nous présentons l'approche locale, le modèle réduit incrémental de rang-1. Dans la troisième section, nous traitons des problèmes de haute dimension souvent rencontrés lors de la conception de voitures : nous considérons l'histoire mécanique, une paramétrisation étendue et une procédure d'apprentissage dans des espaces de haute dimension. Enfin, dans la dernière section, nous nous concentrons sur le schéma d'intégration du modèle réduit dans un solveur de crash.

Le modèle réduit de rang-n

Construire un modèle global d'ordre réduit est une stratégie intéressante pour diminuer le temps de calcul lorsque le domaine des solutions se situe sur une variété de dimension réduite par rapport à l'espace original. Nous avons donc proposé une technique de réduction "a posteriori" pour apprendre un modèle réduit à partir de données de simulation.

Nous proposons de construire une application linéaire de faible rang à partir de données extraites de simulations numériques. Nous appelons cette technique d'apprentissage l'approche de rang-n de l'iDMD [Rei+19].

Dans cette section, que nous ne présentons pas plus en détails dans ce résumé, nous présentons d'abord la construction du modèle dans la phase d'apprentissage. Ensuite,

nous abordons la prédiction et la généralisation sur de nouvelles données.

Le dictionnaire de modèles de rang-1

Construire des approximations locales de l'espace des solutions est une stratégie intéressante pour approcher cet espace lorsqu'il est fortement non-linéaire. En effet, un modèle réduit global augmenterait le temps de calcul ou donnerait lieu à des prédictions inexactes selon la taille de la base réduite. Nous avons proposé une technique de réduction locale en s'appuyant sur l'approche globale pour apprendre un modèle réduit à partir de données de simulation. L'approche locale est adaptée pour prédire le comportement dynamique des points de soudure.

Nous avons développé un dictionnaire d'applications linéaires locales et défini une fonction d'interpolation pour prédire les comportements dynamiques du point de soudure. Nous appelons cette technique d'apprentissage l'approche rang-1 dans l' iDMD [Rei+19]. Cette section présente la construction du modèle et la généralisation à de nouvelles données.

Nous proposons un dictionnaire $\mathfrak{D}$ de $n_{\mathcal{T}}$ modèles linéaires locaux de rang 1 qui approchent localement la variété :

$$\mathfrak{D} = \left\{ \mathcal{K}_{lr}^{1}, ..., \mathcal{K}_{lr}^{n_{\mathcal{T}}} \right\}$$

Chaque modèle local $\mathcal{K}_{lr}^{i}$ est lié à un couple d'instantanés $(\mathbf{U}, \mathbf{F})^{i}, i \in [\![0, n_{\mathcal{T}}]\!]$ extrait de simulations haute-fidélité nécessitant des calculs coûteux. Le vecteur $\mathbf{U}^{i} \in \mathbb{R}^{N_{dof}}$ représente les déplacements extraits de l'instantané à t^{i} et le vecteur $\mathbf{F}^{i} \in \mathbb{R}^{N_{dof}}$ représente les efforts extraits de l'instantané t^{i}.

Pour construire le dictionnaire $\mathfrak{D}$, nous procédons de manière incrémentale. Le premier modèle local est calculé en minimisant :

$$\mathcal{K}_{lr}^{1} = \operatorname{argmin} \left\| \mathbf{F}^{1} - \mathcal{K}_{lr}^{1} \mathbf{U}^{1} \right\|_{F}$$

En utilisant une multiplication à gauche avec le vecteur transposé $\mathbf{F}^{1^{\mathrm{T}}}$ on obtient une matrice symétrique, comme c'est le cas pour les matrices de rigidité, avec l'avantage

d'éviter l'inversion matricielle mal conditionnée de $(\mathbf{U}^1\mathbf{U}^{1^\mathsf{T}})$:

$$\mathcal{K}_{lr}^1 = \frac{\mathbf{F}^1\mathbf{F}^{1^\mathsf{T}}}{\mathbf{F}^{1^\mathsf{T}}\mathbf{U}^1}$$

Lorsque le deuxième couple de données $(\mathbf{F}^2, \mathbf{U}^2)$ est échantillonné, nous construisons également le modèle local associé $\mathcal{K}_{lr}^2$. Nous procédons ainsi pour tout nouveau couple de données $(\mathbf{F}^i, \mathbf{U}^i)$ pour construire le modèle associé $\mathcal{K}_{lr}^i$ et remplir le dictionnaire $\mathfrak{D}$.

Cette construction locale présente quatre avantages principaux. Premièrement, elle ne nécessite pas de calculs itératifs pour obtenir le modèle local. En effet, le résultat analytique de la minimisation permet de calculer le modèle local. Deuxièmement, un simple produit tensoriel est nécessaire pour construire le modèle local. Par conséquent, un dictionnaire $\mathfrak{D}'$ rempli de couples de vecteurs de tailles respectives $(N_{dof} \times 1)$ et $(1 \times N_{dof})$ peut être utilisé, au lieu d'un dictionnaire de matrices de taille $N_{dof} \times N_{dof}$. Troisièmement, ces modèles locaux ont l'avantage de combiner linéairement toutes les composantes de $\mathbf{U}^i$ pour prédire n'importe quelle composante de $\mathbf{F}^i$. Par exemple, un déplacement dans la direction y dans un nœud peut influencer la force dans la direction x dans un autre nœud. Enfin, dès qu'une donnée est échantillonnée, un modèle (localement précis) peut être construit, contrairement à d'autres méthodes de réduction comme la "Proper Orthogonal Decomposition" où plusieurs instantanés doivent être extraits pour construire un modèle.

Dès que le dictionnaire est rempli de modèles (ou d'au moins un modèle), nous pouvons prédire la sortie liée à toute nouvelle entrée $\mathbf{U}$. Pour cela, chaque modèle local $\mathcal{K}_{lr}^i$ dans $\mathfrak{D}$ a été préalablement étiqueté avec l'entrée $\mathbf{U}^i$ utilisée dans sa construction.

$$\mathfrak{D} = \begin{Bmatrix} \mathcal{K}_{lr}^1, & \cdots & , \mathcal{K}_{lr}^{n_\mathcal{T}} \\ \downarrow & & \downarrow \\ \mathbf{U}^1, & \cdots & , \mathbf{U}^{n_\mathcal{T}} \end{Bmatrix}$$

Le modèle employé pour la prédiction est noté $\mathcal{K}(\mathbf{U})$. Il est interpolé à partir des modèles rang-1 dans le dictionnaire $\mathfrak{D}$ selon la formule suivante :

$$\mathcal{K}_{lr}(\mathbf{U}) \approx \sum_{i=1}^{n_\mathcal{T}} \mathcal{K}_{lr}^i \mathcal{I}(\mathbf{U}^i) \tag{1}$$

avec $\mathcal{I}(\mathbf{U}^i)$ la fonction d'interpolation opérant dans l'espace des $\mathbf{U}^i$. Cette fonction est supposée décroissante avec la distance entre $\mathbf{U}$ et $\mathbf{U}^i$. Plusieurs fonctions peuvent être considérées, dans la suite nous utilisons l'interpolation des plus proches voisins.

La prédiction des efforts $\mathbf{F}$ associés aux nouveaux déplacements est obtenue avec le produit matriciel suivant:

$$\mathbf{F} = \mathcal{K}_{lr}(\mathbf{U})\mathbf{U}$$

Les espaces de grande dimension

Les simulations de crash test rencontrent un ensemble de problèmes en haute dimension lors du développement d'un véhicule. En effet, ces simulations mettent en jeu de nombreux degrés de liberté $N_{dof} \gg 1$ et des comportements dynamiques qui dépendent de l'histoire mécanique. De plus, ces simulations sont utilisées dans un cadre de conception multi-objectifs. Ces problèmes à haute dimension sont liés à la "malédiction de la dimmension"[AK18]. Dans cette section, nous présentons des approches visant à résoudre ces problèmes avec l'iDMD.

Pour répondre à ces problèmes, nous avons étendu le domaine paramétrique de l'iDMD en utilisant de nouveaux paramètres. Ces paramètres tiennent compte des différents critères de conception et de l'histoire mécanique. Nous avons également associé l'iDMD à une base réduite pour opérer dans des espaces de haute dimension ($N_{dof} \gg 1$).

Dans cette section, que nous ne détaillons pas dans ce résumé, nous présentons la prise en compte de l'histoire mécanique, puis nous discutons la paramétrisation de l'apprentissage dans un cadre de conception multi-requêtes, et enfin nous introduisons l'iDMD associée aux bases réduites.

La condensation mécanique revisitée

Cette section présente maintenant le couplage du modèle réduit avec le modèle global. Nous nous sommes référés à la condensation statique que nous introduisons dans le chapitre 1.

La condensation statique fait partie des méthodes de sub-modeling. Elle décompose donc le domaine d'intérêt Ω en deux sous-domaines. Le domaine global $\tilde{\Omega}$ où une discrétisation grossière est employée et le domaine local ω finement maillé. Pour éliminer le domaine finement maillé du modèle, les effets du domaine local sur le domaine global sont réexprimés sur la frontière locale-globale $\partial\omega$ qui sépare les deux domaines.

Nous avons formulé les hypothèses suivantes pour réaliser notre couplage :

- A chaque instant t^n, le comportement du domaine local ω peut-être approché par un modèle quasi-statique.
- Tous les efforts appliqués sur le domaine local ω passent par la frontière locale-globale $\partial\omega$. Il n'y a pas d'efforts extérieurs appliqués sur le domaine local.

Selon ces hypothèses, à chaque instant t^n le domaine Ω est donc régi par une équation élasto-statique dont la forme discrète se lit :

$$\mathcal{K}\mathbf{U}^n = \mathbf{F}^n,$$

où $\mathbf{U}$ représente le vecteur qui contient les déplacements nodaux et $\mathbf{F}$ représente les charges externes nodales. Les vecteurs $\mathbf{U}$ et $\mathbf{F}$ peuvent être décomposés en $\mathbf{U} = (\mathbf{U}_g, \mathbf{U}_b, \mathbf{U}_i)$ et $\mathbf{F} = (\mathbf{F}_g, \mathbf{F}_b, \mathbf{F}_i)$ avec :

- $\mathbf{U}_g$ et $\mathbf{F}_g$ les déplacements et les forces aux nœuds du domaine global $\tilde{\Omega}$,
- $\mathbf{U}_b$ et $\mathbf{F}_b$ les déplacements et les forces dans les nœuds de la frontière local-global ω,
- $\mathbf{U}_i$ et $\mathbf{F}_i$ les déplacements et les forces aux nœuds du domaine local ω. Les efforts $\mathbf{F}_i$ sont supposés nuls.

Nous nous concentrons sur les nœuds de la frontière local-global et sur les nœuds internes pour appliquer la condensation statique. Le problème peut être décomposé à l'instant t^n en :

$$\begin{pmatrix} \mathcal{K}_{bb} & \mathcal{K}_{bi} \\ \mathcal{K}_{ib} & \mathcal{K}_{ii} \end{pmatrix} \begin{pmatrix} \mathbf{U}_b^n \\ \mathbf{U}_i^n \end{pmatrix} = \begin{pmatrix} \mathbf{F}_b^n \\ \mathbf{0} \end{pmatrix}. \tag{2}$$

Pour condenser les degrés de liberté internes, nous avons développé la deuxième relation dans l'équation 2 et exprimé les déplacements internes $\mathbf{U}_i$ en utilisant les autres variables. Par conséquent, la première relation de l'équation peut être réécrite de manière plus compacte comme suit :

$$\tilde{\mathcal{K}}_{bb}\mathbf{U}_b = \mathbf{F}_b \tag{3}$$

avec $\tilde{\mathcal{K}}_{bb} = \mathcal{K}_{bb} - \mathcal{K}_{bi}\mathcal{K}_{ii}^{-1}\mathcal{K}_{ib}$

La matrice $\tilde{\mathcal{K}}_{bb}$ peut être facilement apprise à partir de couples de données $(\mathbf{U}_b, \mathbf{F}_b)$ extraits sur la frontière locale-globale $\partial\omega$. L' approche locale de l'iDMD qui construit des modèles locaux linéaire sera utilisée dans la suite pour apprendre le comportement de la frontière locale-globale dans l'application sur les points de soudure présentée dans le chapitre suivant.

Chapitre 3 : application aux points de soudure

En utilisant de l'approche de rang-1 et la procédure locale-globale présentée dans le chapitre 2, nous avons évalué les performances d'un modèle réduit d'un point de soudure méso-échelle couplé à un solveur sur un cas simple. Dans ce chapitre, nous nous intéressons d'abord à la précision et au temps de calcul du modèle réduit en dehors du solveur puis au couplage du modèle réduit avec le solveur en utilisant l'approche locale-globale.

Le modèle réduit construit avec l'approche rang-1 a réduit le temps de calcul de simulations de validation haute-fidélité de 1h30 à moins de 8 minutes sur 16 processeurs. L'erreur relative de prédiction était inférieure à 10% avant la rupture et est restée globalement acceptable pendant la propagation de la fissure. De plus, l'approche locale-

globale a montré des résultats prometteurs dans le solveur. À la suite de l'obtention de ces résultats encourageant, une étude plus approfondie a démarré pour traiter de cas de chargements complexes et de l'industrialisation. Cette étude est conduite par l'équipe recherche de ESI.

Dans la première section de ce chapitre, nous présentons la configuration numérique et la construction de l'ensemble de données d'entraînement. Dans la deuxième section, nous nous concentrons sur la validation des hypothèses concernant la procédure de rang-1. Enfin, nous présentons les performances des modèles rang-1 et l'approche locale-globale dans la dernière section.

Construction du jeu de données

Nous avons défini une expérience numérique pour tester notre approche. La première étape consiste à extraire des données d'entraînement à partir de simulations haute-fidélités afin de construire notre modèle réduit de la frontière.

Nous avons simulé le comportement d'un point de soudure reliant deux tôles lors d'essais unitaires (comme des essais de pelage, de traction, de cisaillement ou mixtes). Pour définir les paramètres de sollicitation des simulations, nous avons utilisé un plan d'expérience de cinq paramètres binaires qui régissaient l'encastrement de la tôle inférieur (deux positions), les nœuds sollicités de la tôle supérieure (deux configurations) et les trois directions de sollicitations (selon deux intensités). Nous avons obtenu 28 simulations d'entraînement haute-fidélité de rupture de points de soudure avec VPS.

En suivant la procédure locale-globale, nous avons décomposé le domaine d'intérêt en deux domaines, le domaine local ω qui contient le point de soudure finement maillé entouré par des éléments coques représentant les tôles et le domaine global $\tilde{\Omega}$ qui contient des élément coques représentant les tôles. La frontière locale-globale $\partial\omega$ se situe donc sur des nœuds d'éléments coques afin de faciliter le couplage.

Nous avons extrait des instantanés des simulations les déplacements et les efforts sur les nœuds de la frontière $\partial\omega$ entre le domaine local et le domaine global pour construire l'ensemble d'entraînement $\mathcal{T}$. Ainsi, le modèle réduit va apprendre les comportements des nœuds de la frontière liés à l'espace de chargement paramétrique défini par le plan

d'expérience.

Validation de l'approche locale

Comme nous l'avons présenté dans le chapitre 1, les comportements dynamiques des points de soudure sont complexes à prédire car ils sont fortement non linéaires, en particulier près du stade de la rupture. Dans cette section que nous ne détaillons pas dans ce résumé, nous avons validé qu'une approche locale était plus pertinente qu'une approche globale pour apprendre le modèle réduit de la frontière.

Nous avons donc choisi la procédure locale de rang-1 pour construire le modèle réduit. Nous avons construit un dictionnaire d'application linéaire locales $\mathfrak{D}_b$ à partir du jeu d'entrainement $\mathcal{T}$ et défini comme fonction d'interpolation pour la phase de prédiction l'interpolation des plus proches voisins.

Prédiction sur le jeu de données test

Nous avons abordé la prédiction de nouveaux comportements de points de soudure qui n'appartiennent pas à l'ensemble d'entraînement $\mathcal{T}$ pour simuler un problème industriel. Ainsi, nous avons défini une nouvelle expérience numérique et évalué les performances de l'approche de rang-1 ainsi que l'intégration locale-globale dans VPS.

Avec un nouveau plan d'expérience, nous avons calculé 6 simulations de test haute-fidélité avec VPS. Les paramètres utilisés pour définir les sollicitations sont les mêmes que pour le jeu d'entrainement mais les intensités de sollicitation ne sont plus des paramètres binaires.

Nous avons comparé la prédiction du comportement de la frontière avec le modèle réduit local (en utilisant le dictionnaire local $\mathfrak{D}_b$) et le comportement des nœuds de la frontière extraits des simulations haute-fidélité. Nous avons constaté que l'erreur de prédiction du modèle réduit était inférieure à 10% avant la rupture et restait acceptable après. De plus, le modèle réduit opère rapidement (8 minutes contre 1h30 pour le modèle méso-échelle). Bien évidemment, le modèle réduit est moins riche car il prédit uniquement le comportement des nœuds de la frontière alors que le modèle méso-échelle prédit l'intégralité du comportement du point de soudure. Cependant,

ce sont les effets du point de soudure sur les tôles qui nous intéressent dans un crash test et non les dommages à l'intérieur du domaine local. À la suite de ces résultats encourageants, un couplage du modèle réduit avec le domaine global a été réalisé pour valider l'approche locale-globale.

En intégrant le modèle réduit selon cette approche, le comportement du domaine global est prédit en approchant la précision que l'on aurait obtenue avec un modèle méso-échelle au le pas de temps critique du domaine global. Ces résultats encourageants ont motivé l'intégration future du modèle réduit local avec l'approche locale-globale dans VPS pour simuler le comportement des points de soudure dans le calcul de crash tests.

Chapitre 4 : conclusions

Le résumé de ce dernier chapitre se concentre sur les conclusions, les limites et les perspectives de notre étude.

Conclusions

Nous avons étudié l'intégration d'un modèle réduit du point de soudure au lieu d'un modèle à méso-échelle dans le solveur VPS afin de diminuer le temps de calcul d'une simulation de crash test.

Le modèle réduit que nous avons proposé a donné des résultats plus rapides que le modèle méso-échelle de référence. La précision du modèle réduit est excellente tant que les dommages ne sont pas avancés. L'erreur relative était d'environ 10% avant la rupture et l'instant de la rupture était également prédit avec précision. Pendant la rupture, l'erreur de prédiction reste faible. Lorsqu'un dictionnaire riche était disponible, nous avons obtenu d'excellents résultats.

Dans cette thèse, nous avons également proposé une méthode de couplage du modèle réduit avec le solveur et évalué cette intégration. Lorsque le modèle réduit est introduit pour modéliser un point de soudure, la simulation peut être effectuée avec les effets du domaine local (du point de soudure) mais sans les calculer explicitement.

Les deux principaux avantages d'une telle procédure sont, premièrement, la réduction du nombre de degrés de liberté puisque le modèle global ne considère plus les discrétisations 3D fines des modèles locaux. Deuxièmement, l'augmentation du pas de temps dans l'intégration explicite. En effet, la stabilité de l'intégration n'est plus gouvernée par la plus petite taille de l'élément utilisé pour décrire la solution au niveau du domaine local.

Limites

Il est encore trop tôt pour conclure sur la précision et la réduction du temps de calcul de l'approche proposée dans un contexte purement industriel. Cette thèse a illustré les performances de l'approche locale-globale et de l'iDMD dans un cas d'utilisation simple. Pour conclure sur son applicabilité industrielle, le modèle doit être enrichi : (i) en élargissant l'échantillonnage utilisé dans l'apprentissage du modèle ; (ii) en considérant une paramétrisation des déplacements pour traiter des chargements non monotones ; et (iii) en mettant en place la méthodologie locale-globale avec une étude fine de la propagation des erreurs dans le modèle global et les économies de temps de calcul.

L'augmentation de la taille du dictionnaire pour modéliser un ensemble vaste de sollicitations pourrait générer des problèmes de mémoire et ralentir la phase de prédiction. En effet, la rapidité de la fonction d'interpolation $\mathcal{I}$ que nous avons utilisée, l'interpolation par le plus proche voisin, dépend de la taille du dictionnaire. Cependant, les ingénieurs conçoivent les voitures pour éviter des modes de rupture spécifiques, donc l'espace des chargements pourrait être d'une taille raisonnable. Nous nous attendons à ce que les performances soient significatives en raison du nombre élevé de points de soudure (des milliers dans une voiture) et de l'évaluation ultra-rapide du modèle réduit construit si la taille du dictionnaire reste acceptable.

Perspectives et travaux en cours

Nous avons défini un plan de travail comprenant plusieurs tâches pour évaluer l'industrialisation de notre approche. Ces tâches seront réalisées par l'équipe recherche de ESI Group. La première tâche consiste à réaliser de nombreuses simulations de crash

tests avec des modèles méso-échelle de points de soudure afin de définir le chargement représentatif des crash tests réels. Cette tâche fournira une estimation de la dimension réelle du problème (à laquelle la taille du dictionnaire est liée) et des chargements typiques que les points de soudure rencontrent dans les crashs tests. La deuxième tâche concerne la validation de l'approche en comparant les résultats des calculs de crash test effectués avec les modèles de points de soudure à méso-échelle avec les mêmes calculs effectués avec les modèles de points de soudure réduits. Il s'agit de vérifier que les modèles réduits n'induisent pas une divergence du calcul. Si les résultats sont concluants, une intégration industrielle dans VPS sera menée, participant à l'amélioration de la précision du calcul afin de réaliser des simulations haute-fidélités.

Remerciements

Je remercie mon directeur de thèse Monsieur Francisco Chinesta, Professeur aux Arts et Métiers, pour son encadrement. J'ai beaucoup apprécié nos échanges scientifiques qui m'ont permis de mener cette thèse à son terme.

Je remercie ma co-directrice de thèse Docteur Fatima Daim, pour son encadrement et son soutien indéfectible dans les moments difficiles. J'ai énormément appris, scientifiquement et humainement, grâce à notre riche collaboration.

Je remercie également Messieurs les Professeurs Antonio Falco et Carl Labergère pour leur intérêt porté à cette thèse et pour avoir accepté d'être rapporteur.

Mes remerciements vont également à Monsieur Fodil Meraghni, Professeur aux Arts et Métiers, Madame Marianne Béringhier, Maître de conférences à l'ISAE ENSMA, Docteur Christine Royer et Docteur Mustapha Ziane pour avoir accepté de participer à ce jury de thèse en tant qu'examinateurs. Je remercie également Docteur Yves Tourbier pour ses conseils et pour avoir accepté de participer à ce jury de thèse.

Merci à mes collègues chez ESI Group et au PIMM pour leur soutien. En particulier, merci au Docteur Gabriel Fougeron pour nos échanges scientifiques, sa relecture rigoureuse et son aide pendant la préparation de la soutenance. Merci à Madame Anne Chambard pour son suivi pendant cette thèse. Merci également au Docteur Nicolas Hascoët pour ses conseils et son aide quotidienne pendant ces trois années.

J'adresse aussi mes remerciements à Madame Claire Mandon, Madame Florence Dumard et Madame Valentine Fredj pour leur aide administrative.

Ce travail n'aurait pas été possible sans le soutien de ESI Group, du laboratoire PIMM

aux Arts et Métiers et de l'École Doctorale Sciences des Métiers de l'Ingénieurs.

Je remercie enfin ma famille et mes amis pour leurs encouragements pendant ces trois années, sans eux la tâche aurait été bien plus difficile.

Contents

List of Tables

List of Figures

Acronyms

AHSS Advanced High Strength Steels. 52, 53

BM Base Metal. 51, 54

CAFE Corporate Average Fuel Economy. 40

CFL Courant-Friedrichs-Lewy. 45

CPU Central Processing Unit. 45, 80, 91, 111

DEIM Discrete Empirical Interpolation Method. 60

DMD Dynamic Mode Decomposition. 60, 61, 66, 67, 117

DoE Design Of Experiment. 26, 57, 90, 93–95, 101, 102, 105–111

EIM Empirical Interpolation Method. 60

EuroNCAP European New Car Assessment Program. 27, 38, 39

FZ Fuzion Zone. 52, 54, 90, 91

HAZ Heat Affected Zone. 51, 54, 90, 91

HPC High-Performance Computing. 8, 41, 46, 56, 116

iDMD incremental Dynamic Mode Decomposition. 2, 4, 10–12, 14, 66, 67, 70, 73, 75, 77, 79, 85, 86, 90, 96, 117, 118

IRS Improved Reduced System. 64

kernel-PCA kernel Principal Component Analysis. 96, 97

local-PCA local Principal Component Analysis. 62, 64

LOOCV Leave-One-Out Cross-Validation. 28, 100, 101

MOR Model Order Reduction. 37, 56, 60, 64, 67, 70, 89, 90, 96, 97, 116, 119

PCA Principal Component Analysis. 62, 96, 97

PDE Partial Differential Equation. 42, 43, 56, 61

PGD Proper Generalized Decomposition. 59, 61, 64, 69

POD Proper Orthogonal Decomposition. 59, 62, 64, 72

ReCUR Regression CUR. 58

ROM Reduced-Order Model. 57–59, 61, 62, 64, 66, 67, 70, 79, 80, 85–87, 89–91, 96, 97, 99–101, 103, 113, 116–119

SVD Singular Value Decomposition. 69

UTAC Union Technique de l'Automobile, du motocycle et du Cycle. 38

VPS Virtual Performance Solution. 8, 17–19, 21, 41–44, 54–56, 89, 90, 101, 102, 111, 113, 117, 119

Nomenclature

α	Thermal diffusivity of the medium
$\ddot{\mathbf{U}}(t)$	Semi-discretized acceleration
$\ddot{\mathbf{U}}^n$	Fully discretized acceleration at t^n
Δt	time step
$\dot{\mathbf{U}}(t)$	Semi-discretized velocity
$\dot{\mathbf{U}}^n$	Fully discretized velocity at t^n
$\dot{\mathbf{u}}_0$	Initial condition on velocity
ϵ	Deformation
$\mathbf{F}^i$	i-th output snapshot
$\mathbb{R}$	Set of real numbers
$\mathbf{C}_j$	j-th column vector
$\mathbf{f}$	External force on part of Ω
$\mathbf{f}^i$	Reduced counterpart of $\mathbf{F}^i$
$\mathbf{g}$	Tensile force on part of Ω

$\mathbf{R}_j$ j-th row vector

$\mathbf{S}$ Fully discretized heat source

$\mathbf{u}$ State variable, displacements

$\mathbf{u}_0$ Initial condition on displacements

$\mathbf{v}^i$ Reduced counterpart of $\mathbf{U}^i$

$\mathcal{B}$ Projection matrix

$\mathcal{E}$ Test set

$\mathcal{F}$ Matrix containing the snapshot $\mathbf{F}^i$ as its i-th column

$\mathcal{I}$ Interpolation function on $\mathbf{U}$

$\mathcal{J}$ Interpolation function on $\boldsymbol{\mu}$

$\mathcal{K}$ Stiffness matrix

$\mathcal{K}_{bb}$ Submatrix of $\mathcal{K}$

$\mathcal{K}_{bi}$ Submatrix of $\mathcal{K}$

$\mathcal{K}_{ib}$ Submatrix of $\mathcal{K}$

$\mathcal{K}_{ii}$ Submatrix of $\mathcal{K}$

$\mathcal{K}_{lr}$ Low rank counterpart of $\mathcal{K}$

$\mathcal{K}_{lr}^i$ Local rank-1 model

$\mathcal{L}$ Learning set

$\mathcal{M}$ Mass matrix

$\mathcal{P}$	Projection matrix
$\mathcal{T}$	Train set
$\mathcal{U}$	Matrix containing the snapshot $\mathbf{U}^i$ as its i-th column
$\mathcal{V}$	Validation set
$\mathfrak{D}$	Dictionary of local rank-1 models
$\mathfrak{D}'$	Dictionary of local tensor couples
$\boldsymbol{\mu}$	vector of input and design parameters
Ω	Continuous domain
ω	local domain
$\partial\Omega$	Boundary of the domain Ω
$\partial\omega$	Local-to-global boundary
σ	Stress tensor
$\mathbf{U}(t)$	Semi-discretized displacements
$\mathbf{U}^i$	i-th input snapshot
$\mathbf{U}^n$	Fully discretized displacements at t^n
$\mathbf{U}_b$	Displacements on boundary degrees of freedom
$\mathbf{U}_i$	Displacements on internal degrees of freedom
C	Courant number
n	Manifold dimension

N_r	Retained degrees of freedom
N_t	Truncated degrees of freedom
$n_{\mathcal{E}}$	number of snapshots in the test set
$n_{\mathcal{T}}$	number of snapshots in the train set $\mathcal{T}$
N_{dof}	Number of degrees of freedom
S	System of partial differential equations governing the dynamic of a continuous domain
T^f	Final simulation time
t^n	Time at $t = n\Delta t$
$\mathbf{s}(x,t)$	Heat source in a rod

Chapter 1

Introduction

Summary

Engineers perform virtual and real crash tests to verify that the vehicle being designed meets the safety requirements. The central objective of car manufacturers is to perform only design confirmation crash tests before release to reduce costs and design time. High-fidelity virtual crashes obtained in less than a day are necessary to achieve this goal. Engineers introduce meso-scale models of small components in the car model to obtain excellent prediction accuracy, but the integration of these local models increases computational time. In this chapter, we discuss the integration of accurate models of spot welds, small joining components, into crash test solvers.

To compute crash simulations, the solver follows a numerical scheme approximating the solution of a mathematical problem representing the physics in a crash. The numerical scheme used for the fast dynamics during car crashes is the explicit scheme. Unfortunately, this scheme induces long computations, sometimes weeks of calculations, when precise and local models, such as spot weld mesoscale models, are integrated. Therefore, we investigated Model Order Reduction (MOR) techniques applied to mesoscale models to reduce high-fidelity crash test simulations computational time.

In the first section, we present the role of virtual crashes inside car project developments. In the second section, we describe the numerical methods inside ESI crash software. In the third section, we focus on the numerical models of spot welds. Finally in the last section, we address model order reduction techniques.

1.1 From crash tests to crash numerical simulation

Passengers' safety and fuel efficiency are central objectives in the automotive industry. Indeed, a car must meet consumption and safety requirements to enter the market. To limit cars' consumption, engineers lighten the vehicles. This strategy is very appealing because it applies both to thermal as to electrical engines. To ensure passengers' safety, the engineers develop protective equipment and reinforce the car structure, which inevitably increases its weight. Faced with tightening consumption restrictions, car manufacturers need high-performance design optimization tools to achieve the optimal weight trade-off.

Engineers have been using crash software to efficiently converge on the best design to meet the specifications. Indeed, virtual tests have extended crashes' initial use for safety certification to design optimization and therefore weight reduction. Car manufacturers' central objective is now to conduct only confirmation crash tests to reduce design costs. Therefore high-fidelity crash simulations are necessary.

In this section we first introduce the safety certification performed with crash tests and then we address the design optimization with crash test simulations.

1.1.1 Certified safety with crash test

A crash test is a destructive test that simulates car accidents according to a pre-defined scenario. Three types of crash tests exist in the automotive industry: internal, homologation, and rating. Car manufacturers perform internal crash tests to evaluate the sturdiness of the car and get feedback. The "Union Technique de l'Automobile, du motocycle et du Cycle" (UTAC) does the homologation tests that are required for car commercialization in Europe. Finally, the European New Car Assessment Program (EuroNCAP) performs independent safety assessments and provides ratings.

The EuroNCAP rating [Eur20a] goes beyond the legal requirements [Eur21b] and manufacturers use it to differentiate a model from its competitors. In 2020, the EuroNCAP tested eleven cars and rated them between three and five stars. A majority of vehicles received five stars [Eur21a], the ultimate grade. During the evaluation, the vehicle must pass a dozen of crash scenarios. The more scenarios passed, the safer a car is.

Those scenarios include, for instance:

- the frontal impact assessment during which the car strikes a wall head-on,
- the frontal small-overlaps assessments during which only a part of the front car strikes a barrier,
- the pole-impact assessment (fig. 1.1).

Figure 1.1 – EuroNCAP pole-impact assessment simulates lateral chocs with trees or poles on a roadside. Image from [Eur20b]. The use case later presented in chapter 3 is related to this scenario.

Engineers follow two strategies to preserve passengers' safety during a crash. First, they define safety areas called survival cells around occupants to limit the invasion of parts during a crash. For instance, the pedal and the wheel must not invade the driver's survival cell. Second, they design safety equipment and the car structure to smooth out the deceleration pulse. Safety equipment includes for instance pre-tension and tension-limitation in seat belts or airbags. The car structure absorbs the energy of the impact through its deformable modules. Those modules experience irreversible plastic deformations, which contribute to smoothing the deceleration pulse during a crash.

This thesis work is related to the second strategy and focuses on the behavior of spot welds. Those are small fastener components that assemble two metal sheets (or more)

by overlapping joints, taking advantage of Joule heating. A car contains, on average, 400 parts. Half of them are in the body in white, the car body frame, which contains from 4000 to 6000 spot welds (fig. 1.2). Their impact during crash tests is significant [RSG95]. Indeed with the inadequate joint, cracks might propagate and weaken the structure during a crash.

Figure 1.2 – A back-end car structure primarily assembled with spot welds from [Don+05].

1.1.2 Car design optimization with virtual crash

The design of lightweight cars is a key objective of the automotive industry, as it helps reduce CO_2 emissions. Moreover, the Corporate Average Fuel Economy (CAFE) in Europe enforces a tax purchase on new cars whose average CO_2 emissions exceeds a specific limit. In 2020 it amounted to 95 € per vehicle per gram of excess.

While reducing emissions calls for lighter cars, ensuring the security of passengers implies structure reinforcements. Finding the optimal design that meets these conflicting needs requires a design optimization process that fits into the project schedule.

Engineering teams use virtual crash test software as a design tool. Virtual crashes help them evaluate different material options, assembly techniques or geometries at a reduced cost. These tests are performed daily during project development and at

different levels, as lightweight design has an impact on all systems. If the prototype does not meet the requirements, engineers propose design modifications called countermeasures. A virtual crash calculation should not take more than one day, otherwise the design optimization will lead to project delays. In the worst cases, engineers cannot integrate the countermeasures. Consequently, unnecessary legacy requirements justified by experience can be maintained, limiting overall performance, especially consumption reduction.

Engineers also perform real world crash tests, but they are expensive as these destructive tests require multiple prototypes to evaluate the different scenarios (frontal, sideways). A crash test costs up to 100k€ in the early design phase and between 10k€ to 20k€ when production begins. Moreover, real world crash tests are always behind the actual vehicle design. Early-stage prototypes tests are costly primarily because of the necessary prototyping tooling and manufacturing delays. Contrary to real world testing, a virtual crash costs a few thousand euros no matter how advanced the project is.

The central objective of car manufacturers is to conduct only design confirmation crash tests to reduce design costs and high-fidelity virtual crashes are needed to achieve this objective. This thesis proposes a new methodology illustrated by a use case on spot welds for better accuracy and fast calculations in virtual crashes.

1.2 Numerical models inside ESI crash software

ESI develops an efficient tool for virtual prototyping called Virtual Performance Solution (VPS) that computes crash simulations. Those simulations are the result of calculations run following a numerical scheme usually on High-Performance Computing (HPC) clusters. This numerical scheme yields a solution to a mathematical problem representing the physics in a crash.

Depending on the desired accuracy, a computation can run for hours, days, and sometimes weeks. However, in an industrial car project, high-fidelity crash computations should not last more than a day to fit into the project schedule.

This section first introduces the mathematical model of the dynamics in a crash, then

focuses on the numerical scheme used in VPS and finally reports the future trends in numerical simulation.

1.2.1 Modeling the dynamics of a continuous medium

Standard simulation in classical mechanics relies on a system (eq. 1.1) of Partial Differential Equations (PDE). The evolution of the body is described by a relationship between the state variable $\mathbf{u}(x, y, z, t)$ (the displacements in a crash simulation) and its spatial and time derivatives. The system usually consists of two sorts of equations: the first related to balance or equilibrium laws and the second to constitutive laws (eq. 1.2). Additional conditions determine the solution—the initial conditions at the beginning of the simulation and the boundary conditions. The resolution of the system S (eq. 1.1) uncovers forces and displacements on a domain Ω, representing the car in a crash simulation.

The dynamics of the continuous domain $\Omega \subset \mathbb{R}^3$ during a simulation time interval $(0, T^f]$ is governed by the following system of partial differential equations:

$$(S) = \begin{cases} \rho \dfrac{\partial^2 \mathbf{u}}{\partial t^2} - \nabla \cdot \sigma &= \mathbf{f} & \text{on } \Omega \\ \sigma.\mathbf{n} &= \mathbf{g} & \text{on } \Gamma_s \times (0, T] \\ \mathbf{u} &= \mathbf{0} & \text{on } \Gamma_u \times (0, T] \\ \mathbf{u}\big|_{t=0} &= \mathbf{u_0} & \\ \dfrac{\partial \mathbf{u}}{\partial t}\Big|_{t=0} &= \dot{\mathbf{u}}_0 & \end{cases} \tag{1.1}$$

With:

- The displacements $\mathbf{u}$,
- The stress tensor σ,
- The divergence $\nabla \cdot \sigma = \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_z}{\partial z}$
- The external force $\mathbf{f}$ applied on Ω
- The tensile force $\mathbf{g}$ applied on part of the boundary of Ω, $\Gamma_S \subset \partial\Omega$.

And the initial and boundary conditions:

- The displacements are zero on a part of the boundary of Ω, $\Gamma_u \subset \partial\Omega$.
- The initial displacements and velocity are respectively $\mathbf{u}_0$ and $\dot{\mathbf{u}}_0$.

The stress tensor σ is a second-order tensor field. It describes the internal forces applied on any virtual surfaces in the medium. Any deformation of a solid material generates internal stress that tends to restore the material to its original state. The constitutive law between stress σ and strain ϵ (deformation) can be intricate and is usually defined from observations. In the system S it follows the law:

$$\sigma = \mathbf{\Sigma}(\epsilon, \dot{\epsilon}) \tag{1.2}$$

Where $\mathbf{\Sigma}$ is the constitutive law.

The strain ϵ represents body deformation. It is the relative change in the position of points within a body that has undergone deformation (rigid-body motions are excluded). With respect to the hypothesis of infinitesimal deformations of a continuum body, it is defined by:

$$\epsilon = \frac{1}{2}(\nabla \mathbf{u}^T + \nabla \mathbf{u})$$

1.2.2 The ESI crash test solver in Virtual Performance Solution

The finite element method is the discrete method used to solve the system S in VPS. In this method, the PDEs in (eq. 1.1) are brought into the weak formulation. A mesh τ_h subdivides the continuous domain Ω into sets of elements. We define on this mesh a finite set of functions called shape functions $\{\varphi_i\}_{i=1}^{N_{dof}}$, with N_{dof} the number of nodes times the number of degrees of freedom in a node. The Galerkin method converts the continuous problem to a semi-discrete problem by multiplying the weak formulation by $\boldsymbol{\varphi}_i$:

$$\int_\Omega \rho \frac{\partial^2 \mathbf{u}}{\partial t^2} \boldsymbol{\varphi}_i + \sigma : \epsilon(\boldsymbol{\varphi}_i) dx = \int_\Omega \mathbf{f} \cdot \boldsymbol{\varphi}_i dx + \int_{\Gamma_S} \mathbf{g} \cdot \boldsymbol{\varphi}_i d\Gamma$$

This equation is then integrated by parts and results in a large system of N_{dof} equations. This semi-discretized system of equations read in matrix form:

$$\mathcal{M}\ddot{\mathbf{U}}(t) + \mathbf{F}_{int}(\mathbf{U}(t),\, \dot{\mathbf{U}}(t)) = \mathbf{F}_{ext}(t)$$

Where $\mathcal{M}$ denotes the mass matrix, $\ddot{\mathbf{U}}(t)$ the acceleration, $\mathbf{F}_{int}$ the internal or cohesion force that depends on the displacements $\mathbf{U}(t)$ and the velocity $\dot{\mathbf{U}}(t)$, and $\mathbf{F}_{ext}(t)$ the external force.

An explicit time scheme is used in VPS to capture the rapid variations of the physical fields as the deceleration in a crash lasts only a few tenths of a second. The calculation scheme starts with an initialization step and continues with an iterative loop (algo. 1). This loop ends when the computation reaches the required number of iterations n_f related to the final simulation time T^f.

Algorithm 1: Conservation of mechanical energy with the leapfrog integration.

for $n \leq n_f$ **do**
- $\mathbf{U}^n = \mathbf{U}^{n-1} + \Delta t^n \dot{\mathbf{U}}^{n-\frac{1}{2}}$;
- $\ddot{\mathbf{U}}^n = M^{-1}\left(\mathbf{F}_{ext} - \mathbf{F}_{int}(\mathbf{U}^n, \dot{\mathbf{U}}^{n-\frac{1}{2}})\right)$;
- $\Delta t^{n+\frac{1}{2}} = \dfrac{\Delta t^n + \Delta t^{n+1}}{2}$;
- $\dot{\mathbf{U}}^{n+\frac{1}{2}} = \dot{\mathbf{U}}^{n-\frac{1}{2}} + \Delta t^{n+\frac{1}{2}} \ddot{\mathbf{U}}^n$;
- $n = n + 1$

end

Where $\mathbf{U}^n$ is the approximation of the displacements at $t = t^n$: $\mathbf{U}^n \sim \mathbf{U}(t^n)$. In the same way we define $\dot{\mathbf{U}}^n \sim \dot{\mathbf{U}}(t^n)$, $\ddot{\mathbf{U}}^n \sim \ddot{\mathbf{U}}(t^n)$ and $t^0 = 0 \leq ... \leq t^{n_f} = T^f$. The time step is denoted by Δt.

A crucial aspect when running numerical simulations is the stability of the time integration scheme, which is a measure of the convergence of a solution. By definition, a scheme is stable if at any time t^n the numerical solution is bounded with respect to a

norm :

$$\|\mathbf{U}^n\| \leq K\|\mathbf{U}^0\|,\ \forall n \geq 0,\ K \in \mathbb{R}^{*+} \tag{1.3}$$

The implicit scheme is usually unconditionally stable, but the previous definition (eq. 1.3) translates into the Courant-Friedrichs-Lewy (CFL) condition for explicit schemes, used for crash simulations. This condition defines the Courant number C [GH07]:

$$C = c_0 \frac{\Delta t}{h} \tag{1.4}$$

Where c_0 denotes an arbitrary constant and h is the size of the smallest element in the mesh τ_h. The convergence is ensured if $C \leq 1$.

When small elements are involved in the mesh, a time step drop is required for computational stability. Hence, a high-fidelity description has a significant impact on the Central Processing Unit (CPU) time. Therefore, the accuracy of crash simulations is limited by the computational time tolerated in vehicle development projects.

1.2.3 Trends in virtual design

High-fidelity crash simulations are a primary goal for car manufacturers, as previously presented in section 1.1.2. The evolution of the number of elements in crash simulation over the last twenty years illustrates the need for high fidelity simulation. In 2005 a car model involved 700 000 elements. In 2010, it rose to 3 million elements and 7.4 million in 2018. High-fidelity crash test simulations become extremely time expensive because of:

- the tiny time steps that usual explicit simulations imply (leading to scalability problems),
- the extremely fine meshes required for adequately describing the different fields.

These issues are not new and do not represent conceptual challenges. They have motivated the proposal of various homogenization procedures, often used when the problems allow separating scales, identifying, and extracting the so-called representative volume elements [Bor08; Fey99; GKB10; Lam+10; YGH09]. Thus, models and solutions

are mature. Predictions are often in agreement with the experimental results [LD18], despite limitations when describing material behaviors under extreme loading conditions.

The main difficulty is definitely the computing time, despite the democratization HPC. Indeed, car manufacturers need to:

- optimize the design regarding the selected quantities of interests (multi-objective optimization),
- perform inverse analysis to calibrate component or global models,
- study the uncertainty propagation that a given deviation around a nominal parameter value will imply in the structural response [Lim+19].

Engineers run many crashes to solve these problems that require high-fidelity simulations. While computing one simulation is acceptable with regards to the computational costs, running hundreds becomes unreasonable.

1.3 Spot welds modeling in virtual crashes

Spot welding is one of the most common assembly processes in a car. As previously introduced, a car contains from 4000 to 6000 spot welds. Engineers use simplified numerical models for spot welds to reduce calculation time in crash simulation. For instance, a crash simulation with simplified spot welds runs for 10 hours on a 140-core computer in 2020 [Gst+20]. However, simplified models can significantly impact the inelastic spot welds' behaviors with the associated effects in the whole structure [Don+05]. Indeed, those models do not consider the intrinsic thermo-mechanical richness of spot welds, involving elasticity, plasticity, damage, and fracture.

High-resolution spot weld models were proposed for improved exactness in response to car manufacturers' need for high-fidelity crash simulations. However, their integration in crash solvers triggers unreasonable computational effort. Thus, we proposed a new methodology in chapter 2 based on model order reduction to circumvent the pre-mentioned issues, and we illustrated this methodology in chapter 3.

In this section we present first the manufacturing process for spot welding. Second, we exhibit the complex, interconnected physical phenomena that models should consider in spot welding. Finally, we focus on the accurate and simplified rupture models and their limits when integrated into crash software.

1.3.1 Spot welding: one of the main assembly processes for cars

Spot welding lies among the oldest types of welding techniques. Elihu Thomson invented the first resistance welding machine at the end of the 19th century in the USA. Quickly after, he developed spot welding that later became the most common resistance welding process widely used in the automotive industry. A spot weld is performed in three operational phases (fig. 1.3): the squeeze phase, the welding phase, and the forging phase. This welding cycle is surrounded by other phases such as loading, unloading the parts, and moving the electrodes forward and backward.

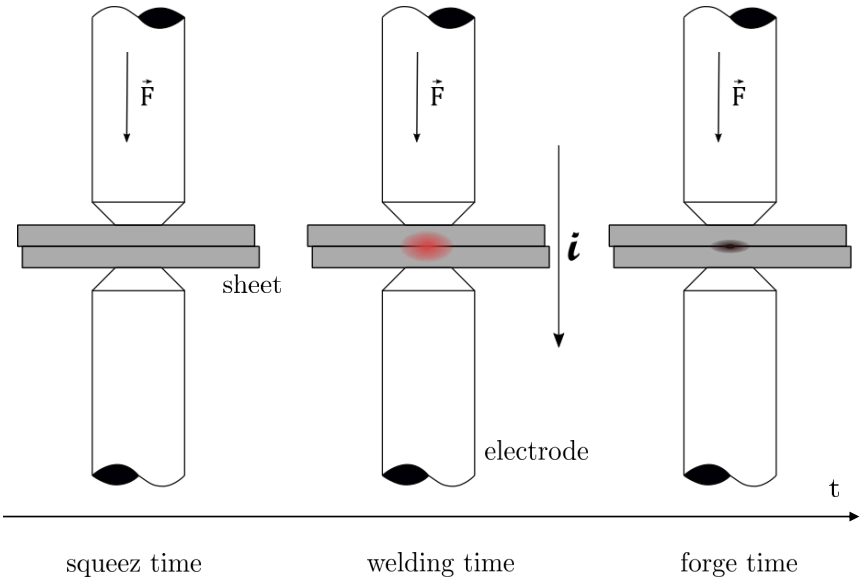


Figure 1.3 – A robot performs a spot weld on two sheets of 1 mm each in 0.20 s.

During the squeeze phase, a pair of mutually opposing electrodes gradually compresses the two metal sheets. The force applied is called the clamping force. Once the electrodes reach a compression threshold, the second phase starts.

During the welding phase, a high-intensity current (between 10^3A to 10^6A [Caz93]) flows through the two electrodes that hold the sheets together. The temperature rises locally because of the short circuit, and the metal melts. Indeed, the current crosses a global electrical resistance R_{eq} (which models the resistance of the electrodes and of the clamped sheets) which causes a heat release due to the Joule effect. The generated heat $\mathcal{Q}$ depends on the current intensity i, the global equivalent resistance crossed $R_{eq}(t)$ and operation time τ:

$$\mathcal{Q} = \int_0^\tau R_{eq}(t) i_2^2(t)\,\mathrm{d}t$$

A liquid nugget then forms around 1000°C in the separation joint between the sheets (fig. 1.3). We present in detail the physical phenomena involved during the heating process in the appendix.

During the forging phase, the electrodes hold the sheets compressed while they cool. Mechanical pressure generates changes in the metal during this operation. It reduces residual stress (due to time differences in cooling) and eliminates bubbles formed by dissolved gases during heating. In addition, the mechanical pressure keeps the liquid nugget between the metal sheets while it solidifies. This prevents expulsion, *i.e.* the ejection of molten metal into the separation joint, which weakens the joint. Once the weld nugget has solidified, the electrodes move to the next welding position.

After cooling and solidification, the spot weld is complete and locally joins the sheets together. Manufacturing robots perform lines of spot welds to join two parts in the automotive industry (fig. 1.4) because it is a discontinuous manufacturing process.

Automotive manufacturers are minimizing the number of spot welds in cars to reduce production costs. Indeed, diminishing operation time is a crucial perspective in lowering production costs [Caz97]. It can be achieved by reducing the number of spot welds to decrease :

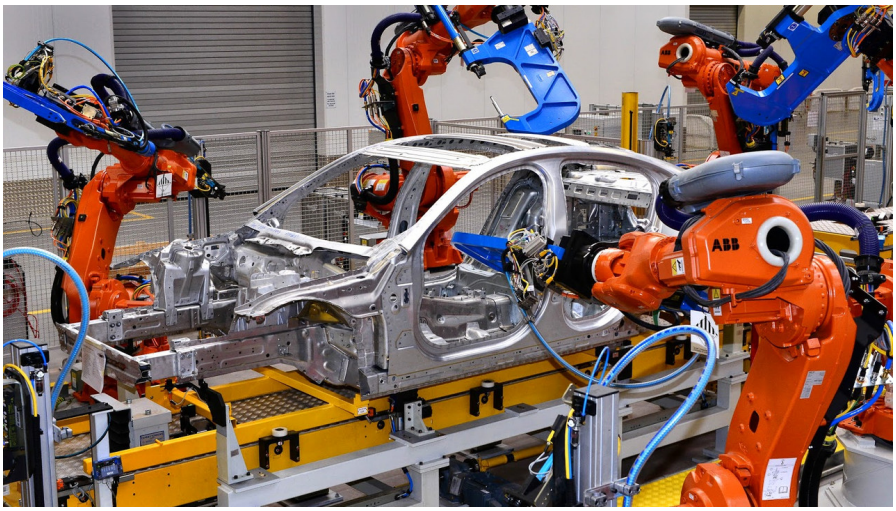


Figure 1.4 – Spot welding robots (in orange) outfitted with a spot-welding gun connects parts by overlapping joint [Dun17]. Some areas are obviously not accessible to the robot.

- the total length of the operation path,
- the number of maintenance operations on electrodes (that lead to production shutdowns).

To achieve this design objective car manufacturers need high-fidelity virtual crashes that represent accurate spot weld behaviors.

1.3.2 Spot welds, complex fastening elements

The spot weld's dynamical behavior depends on weld's geometry and weld's microstructure[1]. This microstructure is modified during spot welding because the metal endures compression, rapid heating and cooling throughout spot welding. Thus, extensive geometrical and metallurgical transformations take place, respectively visible at the small and microscopic scale in the matter. These transformations greatly modify the mechanical properties of the assembly around the spot weld.

Geometry has a significant influence on the mechanical strength of spot welds. For example, Pouranvari et al. found that for shear tensile tests, the size of the nugget is

[1]The microstructure describes the matter at a microscopic scale ($\sim 10\mu$m)

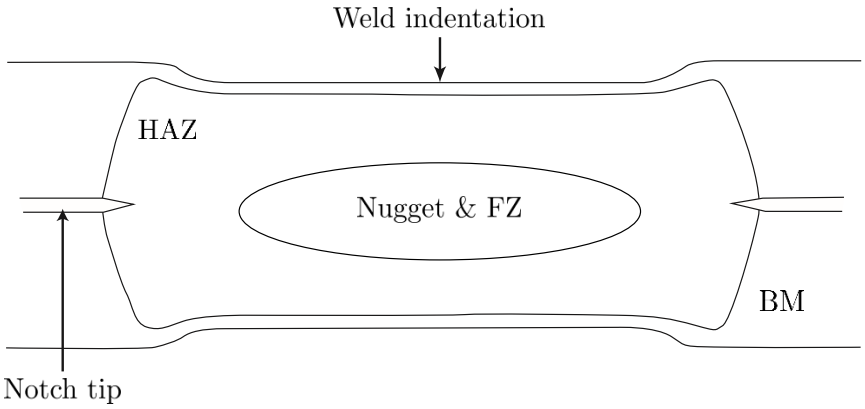


Figure 1.5 – A spot weld is usually symmetric regarding the separating joint. However, asymmetries can appear with heterogeneous assemblies and electrodes misalignment. Significant microstructure transformations take place during welding: three main microstructure zones are usually distinguished in a spot weld.

an important parameter in determining the stress distribution in the weld nugget interface [Pou+07]. Additionally, manufacturing defects often result in geometric defects such as bumps. They influence the origin and propagation of cracks in sheets [LLP06; WP05]. Thus, to capture the precise behaviors of spot welds in crash, those geometric characteristics must be considered in the welding model. Three main geometrical characteristics (fig. 1.5) are usually considered:

- The weld indentation; *i.e* the circular imprint left by the electrodes on the parts.
- The notch tip; *i.e* the small gap between the sheets around the spot weld because the parts are not perfectly flat.
- The welded nugget; *i.e* where metal has reached fusion between the two plates. It is usually of the order of magnitude of the thinnest part in the assembly [Mic16], around 3mm.

Temperature and carbon content[2] variations induce transformation in steel microstructure(fig. 1.6). These transformations modify the sheets' mechanical properties (such

[2]Carbon content is the quantity of carbon blended with iron in steel. It is measured in percentage of total mass

as the strength[3] or toughness[4]). Unalloyed steel near thermodynamic equilibrium at room temperature contains, in a simplified way, two phases called cementite and ferrite. The cementite is composed of iron carbide $\mathrm{Fe_3C}$. The ferrite or α-iron is composed of iron crystals [5] and carbon in a solid solution. Near thermodynamic equilibrium, iron crystal can have two structures:

- Ferrite or α-iron appears below 727°C. It has a body-centered cubic crystal structure.
- Austenite or γ-iron appears between 727°C and 1400°C depending on the carbon content in the steel. It has a face-centered cubic crystal structure.

Carbon atoms diffuse into the steel during these allotropic transformations[6]. When fast cooling γ-iron outside of equilibrium, other crystallographic forms of iron appear, called meta-stable forms. These are hard phases of steel [BHS94] as they are over-saturated with carbon. Indeed, the atoms do not have time to diffuse out of the crystal structure to form cementite. For instance, when cooling at a high rate (around 40–120 °C/s), γ-iron transforms into martensite, a body-centered tetragonal form of iron. Transformations far from equilibrium take place during spot welding as the heating and cooling speed are around 1000 °C/s in the sheets and generate microstructure transformations.

Three main microstructure zones should be considered when trying to predict accurate behaviors in a spot weld [LR08; Ma+08](fig. 1.5):

- In the Base Metal (BM), the temperature rise did not exceed 600°C. Microstructure and mechanical properties are unchanged.
- In the Heat Affected Zone (HAZ), the temperature rise varies between 600°C and

[3]The strength of a material is its ability to withstand an applied load without failure or irreversible deformation.

[4]The material toughness is the amount of energy per unit volume that a material can absorb before rupturing.

[5]The matter in a crystal follows an arrangement called a lattice, which is reproduced regularly. The amorphous matter, however, is highly shambolic. For instance, crystal carbon is diamond, and amorphous carbon is graphite.

[6]Allotropic transformation is the transformation on heating or cooling of a crystalline variety of a metal or alloy into another crystalline variety, from austenite to ferrite for instance

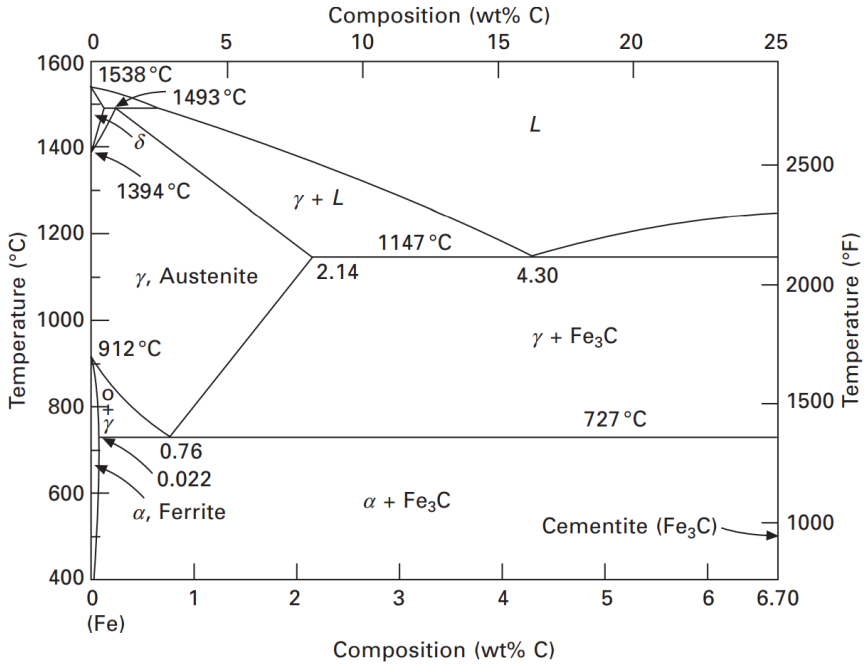


Figure 1.6 – Beginning and ending transformation temperature vary given carbon content near thermodynamic equilibrium [Enn14]. Several crystal phases may coexist in a single steel part, such as austenite and ferrite between 727°C and 912°C for steel below $0,77\%$C.

the fusion temperature (around 1150 °C). A variety of microstructure changes appear depending on carbon content, alloy blend in, and temperature gradient.

- In the Fuzion Zone (FZ), the temperature rise (generally 1500°C) is higher than the fusion temperature. The welded nugget is nearly full of martensite.

The complexity of the transformations during welding increases with alloy mixtures and heat treatments, which is the case for steels used in the automotive industry. For instance, Advanced High Strength Steels (AHSS) are developed to reduce CO_2 emissions and enhance passenger security [KKW08; MY11]. However, those hard steels require accurate models to represent their failure modes in crash simulations [Dum+17].

1.3.3 Spot welds macroscale and mesoscale models for crash

When a spot weld ruptures in an assembly, the mechanical constraints that spread over the remaining ones increase the risk of crack propagation weakening the car structure. Thus, the spot welds are critical points during a car crash, and their failure behaviors need to be taken into consideration to optimize the vehicle design [See+05]. Hence, they should be considered in high-fidelity crash test simulations.

Destructive experimental characterizations have been carried out using test pieces of different shapes [Lin+02; Mah07; ZS97] to understand and model the rupture of welded components. Spot weld rupture usually fails in a combination of two distinct modes [Cha02; Cha03]: the interfacial fracture and the pullout. In an interfacial fracture, the crack propagates through the weld nugget in the separating joint (fig. 1.7b). In the pullout, the weld nugget is pulled from one sheet of metal, leaving a circular hole on the opposite sheet (fig. 1.7a). Interfacial fracture has a lower load-carrying capacity, which is why automotive industry standards [Aut12] are often designed to prevent its occurrence [Wan+06].

(a) Pullout

(b) Interfacial fracture

Figure 1.7 – Spot weld failures are the combination of two main rupture modes.

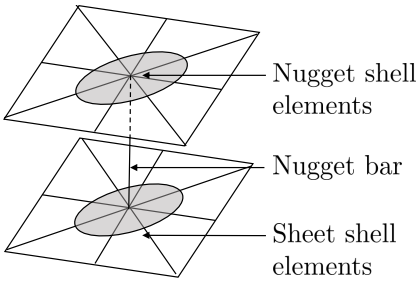
Two types of spot weld models emerge from the literature: the macroscale and the mesoscale models [XD04]. Macroscale models simplify the geometry and mechanical behaviors of spot welds for computational purposes. Simple models focus on elasto-plastic behaviors, and a failure criterion defines rupture[Dum+17]. Therefore, they cannot predict highly non-linear behaviors (fig. 1.8b), including fracture and crack propagation specially for AHSS [Dum+17] but they do not increase the computation time of the crash simulation.

Engineers usually use the two-point rigid element model in the automotive industry [KT14]. This model simulates the welded nugget with a bar element [DG00; Fan+00], rigid or elastic, connecting the upper and lower sheets (fig. 1.8a). It does not represent the HAZ and the BM. The sheets are represented with shell elements [XD04]. The single beam model is relatively inaccurate, mainly because it cannot be satisfactorily updated using material or geometrical parameters [Pal+05].

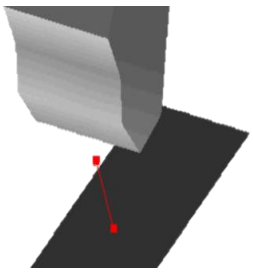
Similar models were developed from the beam approach [KT14]. They involve numerous rigid beams [SVV00; She93] combined with local mesh refinement [XD04]. The beams can follow multiple arrangement such as spider [KT14] or umbrella pattern [Din+06; ZT01]. Analytical solutions in sub-domains can be implemented to increase the accuracy of the finite element model [Viv+02]. However, this method implies pre- and post-processing of numerical data, which is not conceivable in the car industry.

Mesoscale spot weld models are, in contrast, accurate representations [Mah07] but use many elements, typically shell elements [CD00] or 3D solid elements [Dan09; Lam07; PC95; SMB08]. These models consider the geometry of the joint, the mechanical properties of the HAZ, FZ and materials to accurately predict the behavior of spot welds (fig. 1.8).

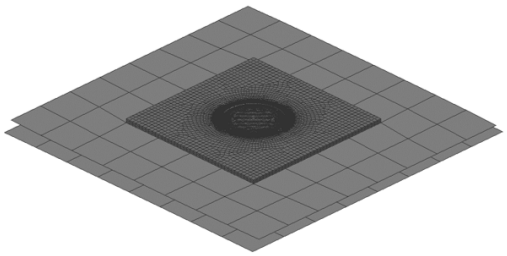
The high-fidelity spot weld model in VPS [Dum+17] accounts for mechanical properties obtained from sheet metal specifications and manufacturing parameters. Finite elements of volume represent the distribution of the three phases (BM, HAZ, FZ) and consider the plastic strain state after welding (fig. 1.8). This high-fidelity model facilitates the integration of new materials for lightweight car design. Indeed, engineers usually follow a tedious weld failure program to calibrate macroscale models in a crash. For instance, Renault's failure program includes 5 iterations of experimental tests for several load conditions to be repeated on each material and thickness combination. As a result, engineers can investigate only a few combinations in terms of test time and cost, which limits the integration of new materials for lighter components [Kah+14]. In contrast, VPS high-fidelity model allows for numerical testing in crashes, as the mechanical properties of spot welds are interpolated from the characteristics of the assembly.



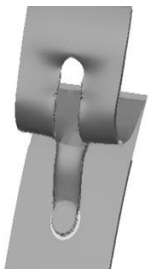
(a) Single beam model



(b) Prediction of the macroscale model



(c) Mesoscale model in VPS



(d) Prediction of the mesoscale model

Figure 1.8 – Macroscale models simplify spot welds behaviors while accurate mesoscale spot weld models can predict cracks propagation.

The integration of mesoscale models in car crashes faces two major difficulties mentioned previously in section 1.2.3. First, since mesoscale models involve numerous elements, the number of degrees of freedom N_{dof} increases tremendously. For instance, the high-fidelity spot weld model of VPS uses 150 000 elements. A standard crash with simplified spot weld requires around 10 million elements and takes 10 hours on HPC clusters. Thus, representing every spot weld with a mesoscale model would require about $10^7 + 4 \times 10^3 \times 10^5$ elements, which gives about 410 million elements. This computation is not feasible on available HPC clusters. Indeed, crash models do not ensure good scalability, and crash solvers that handle this size of problems do not exist. Second, integrating tiny elements in the finite element mesh leads to a time drop for computational stability, increasing computational time. For instance, a crash simulation with 30 mesoscale spot weld models runs for more than three weeks.

From those perspectives, Dumon et al. suggested a methodology that locally integrates accurate models in crash solely where they are critical[Dum+17], *i.e.*, where spot welds rupture. Indeed, the accuracy gap is not significant between macroscale and mesoscale models when the loading induces elastic behaviors [Vim08]. However, this methodology still involves time-consuming iterations. First, engineers run a crash. Then, they integrate accurate models for critical spot welds, and finally, they run a second crash.

Even if solving one case is acceptable regarding the computational cost, solving hundreds of them becomes unreasonable because of the excessive computational resources and processing time required. Hence, the Model Order Reduction (MOR) techniques seem an attractive way forward and are presented in the next section.

1.4 Model order reduction

The system S of PDEs governing the dynamics of the continuous domain Ω (eq. 1.1) considers, in theory, any loading in a vector space of dimension N_{dof}. In practice, however, the system does not often experience all the loading possibilities. Therefore, the solution of a large-scale discrete system belongs to a manifold whose dimension is lower than the number of degrees of freedom of the system. Moreover, the loading usually contains several correlations [Amm+06]. Model order reduction techniques

exploit these powerful but straightforward observations. They assume the solution $\mathbf{U}$ can be defined on a subspace of dimension n, with $n \ll \mathtt{N}_{dof}$.

A Reduced-Order Model (ROM) provides a low-dimensional representation of high-dimensional models, such as finite element models, with comparable accuracy but more rapidly and on a reduced set of inputs, by exploiting correlations in the loading for instance. The resulting reduced model can replace a component represented with an extensive system of equations in a global structure [ASG01] to reduce the number of degrees of freedom and withdraw fine discretizations. This approach is later applied in chapter 3 to integrate a ROM of a spot weld in a crash model. This integration approaches the accuracy of a 3D model at the time step of the global structure.

1.4.1 From surrogate models to reduced-order models

Engineers usually construct simplified models (fig. 1.9) that link the design parameters to the outputs of interest to retrieve optimal design parameters values. Indeed, evaluating all the design possibilities using crash simulations is not practicable regarding the car project schedule.

Those simplified models are called surrogate models [DCE17] or response surfaces. They evaluate design constraints as a function of the numerous design parameters. Indeed, a crash simulation can have up to hundreds of parameters [Gst+20] that comprise part thicknesses or materials.

The construction of a surrogate model unfolds in two steps; first, engineers run several crash simulations for different values of design parameter. Those values are defined with a Design Of Experiment (DoE). Multiple sampling strategies exist to provide the most adequate DoE [BTG94; Kor+12]. The response surface then estimates the output of interest for parameter values that do not belong to the DoE using appropriate interpolation techniques, Kriging being one of the most widely used [Sim+01].

Constructing a surrogate model is a time-consuming task that turns out impractical in high-dimensional spaces [BAC19] when the number of parameters increases, as in the multi-query car design optimization process. Indeed, the model construction requires running several simulations, from 3 to 10 times the number of design parameters con-

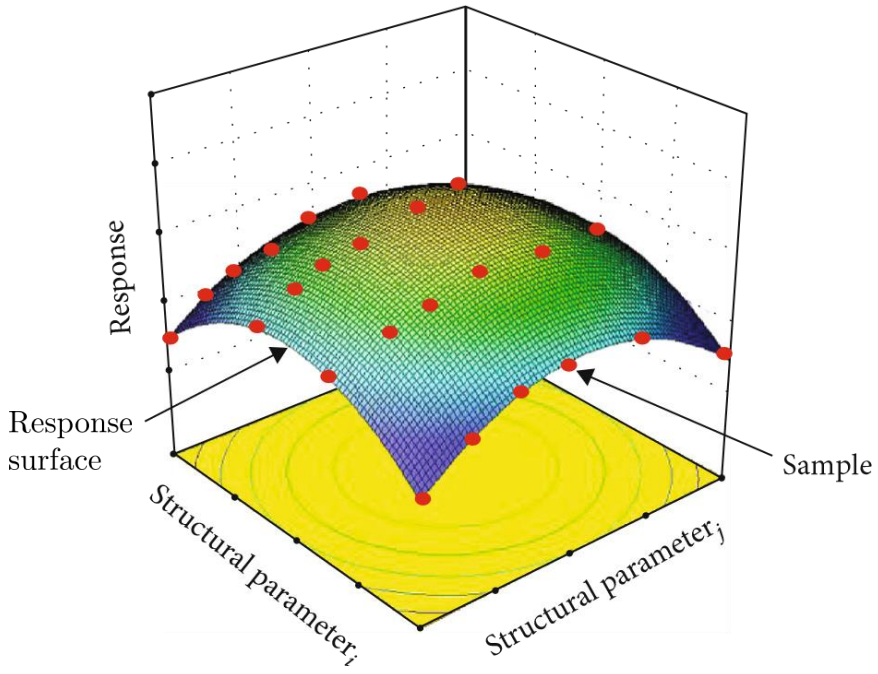


Figure 1.9 – A response surface representing the explicit relationship between the response value and the structural parameters [Hua+20]

sidered in the optimization problem. For instance, a surface response that involves 100 design parameters may be constructed on 1000 crash simulations.

This thesis focuses on model order reduction techniques [Chi+17] that can speed up crash test simulations [Ker+13; Rei+21] and the design optimization process, such as the Regression CUR method (ReCUR, [Gst20; Le +18]). Of course, ROMs also require heavy computations but these techniques are interesting when computing the reduced model is less time-consuming than solving the fine-scale problem for an extensive range of input parameters.

Reduction techniques arise from two main approaches, ”a priori” and ”a posteriori” ROMs. [BAC19; Gst+20]. Both restrict the domain of the solution either through projection techniques over a reduced basis of suitable functions or interpolation techniques. The ”a priori” ROMs are computed by setting an optimization problem. They do not reduce the system of equations S but produce the solution approximation. The

"a posteriori" ROMs are learned from simulation data. They involve a reduced basis extracted from a dataset [ASG01]. This dataset contains high-fidelity simulations carried out offline in the learning stage for different parameter values. Thus, a priori ROMs require offline inspection of the parametric domain. Two methods are detailed below as examples to illustrate these approaches.

The Proper Generalized Decomposition (PGD) constitutes an efficient "a priori" approach. This method relies on a space and time separation of the state variable [CAC10a]. Thus, the PGD approximation is a sum of D functional products involving the time functions $\mathcal{T}_i(t)$ and space functions $\mathcal{X}_i(\mathbf{x})$ that are unknown a priori. The solution in (eq. 1.1) reads in the PGD framework:

$$\mathbf{u}(\mathbf{x}, t) \approx \sum_{i=1}^{D} \mathcal{T}_i(t)\mathcal{X}_i(\mathbf{x}), \tag{1.5}$$

A simultaneous searching of space and time functions was proposed to speed up the procedure [Amm+06; Amm+07; CKL13; CGA16; KO16]. Both groups of functions, $\mathcal{T}_i$ and $\mathcal{X}_i$, are obtained by:

- injecting the approximation (eq. 1.5) into the problem weak form,
- using a greedy algorithm (usually a fixed-point iteration method) for incrementally constructing the separated functions.

This procedure was then extended to address space separation [BLC14], sparse function identification [Ibá+18] and parametric solutions [CAC10a; CAC10b; Chi+13; DZH17] in which geometrical, material and process parameters are assimilated as extra-coordinates. Hence, the resulting model is only solved once to obtain a general solution that includes all the state evolution for every possible value of the parameters. However, the implementation of an integrator is more appealing compared to a parametric solution, as provide by the PGD, regarding the coding implementation in an industrial crash test solver.

The Proper Orthogonal Decomposition (POD) is an "a posteriori" approach. This method provides basis vectors that render simplified evolution of the unknown field [Cha00; Loè60; Ryc+06]. In the snapshot POD introduced by Sirovich [BSK91], the basis vectors

are computed empirically using sampled data extracted from simulations. Snapshots are solutions computed at different instants in time. Here, $\mathbf{U}^i = \mathbf{u}(\mathbf{x}, t^i) \in \mathbb{R}^{N_{dof}}, i \in 1, ..., n_{\mathcal{T}}$ denotes the i^{th} snapshot extracted at $t = t^i$ and $n_{\mathcal{T}}$ is the number of snapshots extracted. The snapshots are stored in a matrix $\mathcal{U} \in \mathbb{R}^{N_{dof} \times n_{\mathcal{T}}}$ containing the snapshots $\mathbf{U}^i$ as its i-th column. The basis vectors are computed by solving the eigenproblem of the covariance matrix $\mathcal{C}$:

$$\mathcal{C} = \frac{1}{\mathtt{n}_s - 1} \mathcal{U}^T \mathcal{U}$$

The basis vectors that correspond to the highest eigenvalues describe the significant modes involved in the simulation. The reduced solution is finally expressed by a sum of eigenvectors $\mathcal{R}_i(\mathbf{x})$ modulated by time coefficients $\gamma_i(t)$ so that:

$$\mathbf{u}(\mathbf{x}, t) \approx \sum_{i=1}^{\mathtt{n}} \gamma_i(t) \mathcal{R}_i(\mathbf{x})$$

whose matrix form reads

$$\mathcal{U} = \mathcal{B}\boldsymbol{\gamma}.$$

The columns of matrix $\mathcal{B}$ contain the reduced functions at node locations. Thus, the component $\mathcal{B}_{ij} = \mathcal{R}_j(\mathbf{x}_i)$ with $i = 1, \ldots, N_{dof}$ and $j = 1, ..., n$. This MOR technique is appealing for linear dimension reduction when we can adequately sample the unknown field.

Other methods have been proposed, such as the hyperreduction also referred to as operator compression [Ryc09; Ryc+06]. Those methods include the Empirical Interpolation Method (EIM), [Bar+04]) and its discrete counterpart the Discrete Empirical Interpolation Method (DEIM, [CS10]). The Dynamic Mode Decomposition (DMD, [Sch10]) is also a data-driven technique that identifies low dimensional time-invariant dynamics. It constructs a linear mapping $\mathcal{A}$ between the snapshots $\mathbf{U}^i$ and $\mathbf{U}^{i+1}$:

$$\mathbf{U}^{i+1} = \mathcal{A}\mathbf{U}^i$$

There are two approaches developed in the DMD to approximate the matrix $\mathcal{A}$. Both

estimate the matrix $\mathcal{A}$ eigenvalues and eigenvectors, referred to as dynamic modes. We present in chapter 2 the construction of a ROM inspired from the DMD and discuss two approaches, a global and a local learning. We introduce the state of the art on local learning in the next section.

1.4.2 Local reduction methods

Sometimes the solution of the governing PDEs lies on a lower-dimensional subspace that a global reduced-order model cannot cover. Therefore, approximating the solution of interest using a global basis can either increase computational time [Row00], or result in inaccurate simulations [AZF12] depending on the size of the reduced-order basis. Local reduced-order models were introduced to address this problem. These models are suited to reduce physical phenomena characterized by different mechanical regimes and parameters variations [AZF12]. For instance, the spot welds mechanical behaviors are very sensitive to the loading history. Therefore, we investigated local methods.

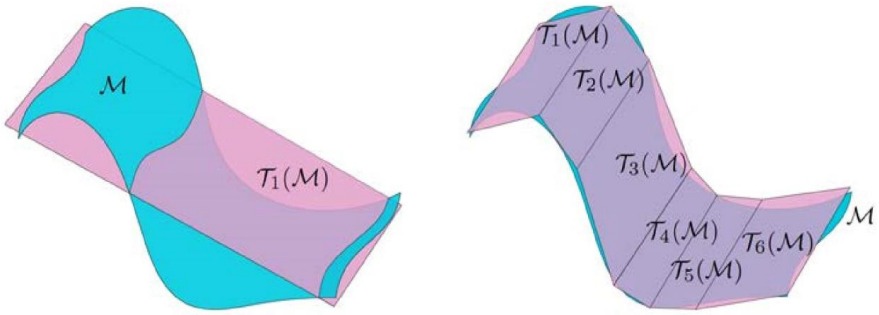


Figure 1.10 – Surface $\mathcal{M}$ where the solution lies and tangent planes $\tau(\mathcal{M})$ where the solution is projected. A global approximation is performed on the left while local reduced models approximate the solution on the right [Bad+17]

Local methods locally approximate the solution in a lower-dimensional subspace generated by an appropriate local basis instead of a global lower-dimensional basis (fig. 1.10). Various local approaches exist, some are based on the interpolation of local reduced-order models [AF08; Was+12] or on the PGD [Bad+17; Ibá+19].

Local methods have been applied to construct reduced order models in various physical fields. Parente et al. [Par+09] have proposed the local Principal Component Analysis [KL97] approach to build a reduced model of a combustion problem. The local-PCA is the local extension of the Principal Component Analysis (PCA), an equivalent method to the POD. Similarly, Washabaugh et al. [Was+12] have applied the concept of reduced-order basis developed in [AZF12] to computational fluid dynamics problems.

Local reduced-order models which require partitioning the input set into clusters can be organized as dictionaries. We introduce in the chapter 2 a dictionary base reduce order model. Dictionary methods appeared in the community of signal processing, and machine learning [AEB06; MBP12]. These methods approximate the solution using an interpolation of few elements from a dictionary that can either be learned or predefined. Thus, the dictionary depends on what the state evolution can be, or on basis functions such as the discrete cosine transform [ANR74]. A dictionary of reduced models as defined in [Dan+20] contains a set of local reduced models linked to sub-regions or clusters using an injective function.

Local methods usually unfold as follow [AZF12; Dan+20]:

- a set of high-fidelity solutions is computed offline,
- this set is partitioned into several clusters offline,
- each cluster is used to build a specific reduced-order model offline,
- the reduced-order basis related to the solution current state is adapted online by finding the closest cluster center

Inputs belonging to the same cluster lead to solutions that are predicted with the same local reduced-order model. Local methods are appealing to reduce the highly nonlinear dynamical behaviors of spot welds. In chapter 2, we introduce a local method to reduce a mesoscale spot weld model.

We investigate in the next section existing integration methods of models into finite element models. Inspired by these approaches, we propose in chapter 2 an integration technique of a ROM in a crash solver.

1.4.3 Local-global approach

Sub-modeling or substructuring approaches have typically been used to diminish the number of degrees of freedom N_{dof} in a finite-element model to reduce computational effort [MH+04]. In particular, the global-local approach refers to the integration of local models in a global model. The local model usually consists of components whose introduction would lead to an excessive number of elements or a time-step drop. The global model then consists of all the remaining components. This approach is appealing to model spot welds in car because of the difference in size.

The integration of sub-models in a global structure reduces the computational effort in two ways, by:

- increasing the characteristic time step Δt
- reducing the number of degrees of freedom N_{dof} in the model

Those techniques often use a projection matrix $\mathcal{P}$, sometimes referred to as a condensation or expansion matrix, to link retained degrees of freedom to truncated degrees of freedom. Thus, the sampled snapshot $\mathbf{U}^i$ is partitioned into the retained degrees of freedom $\mathbf{U}_r^i \in \mathbb{R}^{N_r}$ with N_r the number of retained degrees of freedom and the truncated degrees of freedom $\mathbf{U}_t^i \in \mathbb{R}^{N^t} : N_{dof} = N_r + N_t$. Therefore, $\mathbf{U}^i$ can be expressed as :

$$\mathbf{U}^i = \begin{pmatrix} \mathbf{U}_t^i \\ \mathbf{U}_r^i \end{pmatrix} = \mathcal{P}\mathbf{U}_r^i$$

The form and calculation of the projection matrix $\mathcal{P}$ depend on the method implemented. Some do not have clearly defined projection matrices, such as uncoupled global-local analysis and mass element approximation [MH+04].

Static condensation, sometimes referred to as Guyan reduction, is a popular truncation method in finite element solvers. The static condensation algorithm [Wil74] reduces the finite element model by condensing (truncating) internal degrees of freedom with a Schur complement method. Specifically, it removes all the degrees of freedom not located on the substructure boundary called the local-to-global interface. The remaining degrees of freedom on the boundary retain the stiffness of the local structure but omit the inertial terms. This more compact and consequently more efficient repre-

sentation is exact for static events and the error increases as the effects of the inertia terms become more significant, *i.e.*, for non-static loading conditions. O'Callaghan introduced the Improved Reduced System (IRS) method to improve the static reduction. This method includes the inertia terms as pseudo-static forces. Though the accuracy is improved, it is obviously impossible to emulate the behavior of the entire system.

Dynamic reduction attempts to accurately represent the system dynamics for specific modes that represent the dynamic properties of structures in the frequency domain. The most popular methods include the Craig-Bampton, Dual Craig-Bampton, and Craig-Mayes methods. A comparison of these methods is proposed in [Roe+16] and [Wu18].

Several works have proposed the integration of a MOR in a global model to reduce computational time. Kerfrident et al. [KPB12; Ker+13] developed a local-global framework for fracture. Contrary to the previous approaches, their strategy involves the reduction of the global domain while the local domain where the fracture is experienced is fully solved. The global domain is reduced using the POD, and the coupling technique has strong links with the Craig-Bampton method. Alternatively, Ladevèze et al. introduced a multiscale computational method associated with a reduction using the PGD [LPN10]. Concomitantly, Coussement et al. [CGP13] constructed an integration framework of a reduced model generated by local-PCA into a flow solver. Inspired by these approaches, we propose in chapter 2 an integration scheme for a ROM predicting the highly non-linear behaviors of spot welds.

Chapter 2

Model learning and integration

Summary

Model order reduction of a mesoscale spot weld model is an appealing strategy to reduce computational time of high-fidelity crash simulations. We proposed a novel reduction technique, the incremental Dynamic Mode Decomposition (iDMD) and an integration framework to couple a ROM into a crash solver. This chapter reports the ROM construction and the prediction phase integrated in a larger solver.

Inspired from the DMD, we introduced two data-driven constructors that we published in [Rei+19]. The first is a global linear constructor, while the second is a local linear constructor. Revisiting mechanical condensation, we also provided the ROM integration framework into the crash solver.

In the first section, we introduce first the global reduced model learning procedure, the rank-n constructor. In the second section, we present the local approach, the progressive rank-1 constructor. In the third section, we treat high-dimensional issues often encountered in car design: we consider the mechanical history, the parametrization and a learning procedure in high-dimensional spaces. Finally in the last section, we focus on the integration scheme of the ROM into a crash solver.

2.1 Global learning procedure: rank-n constructor

We previously discussed (in sec. 1.4) that a global reduced-order model is an appealing strategy to reduce computational time when the solution space is embedded in a manifold of a reduced dimension compared to the original space. Hence, we proposed an "a posteriori" MOR technique to learn a ROM from simulation data.

Inspired by the DMD, we developed a linear low-rank mapping relating inputs to outputs extracted from a numerical simulation. We call this learning technique the rank-n constructor in the iDMD [Rei+19] framework.

In this section we report first the model construction in the learning phase. Second, we address the prediction and the generalization on new data.

2.1.1 Model construction in the learning phase

We considered a discrete linear mechanical problem in $\Omega \subset \mathbb{R}^{N_{dof}}$:

$$\mathcal{K}\mathbf{U} = \mathbf{F} \tag{2.1}$$

where $\mathbf{U}$ and $\mathbf{F}$ are respectively displacements and forces vectors while $\mathcal{K} \in \mathbb{R}^{N_{dof} \times N_{dof}}$ represents the discrete model. In a more general framework $\mathbf{F}$ and $\mathbf{U}$ represent the input and output vectors respectively. Their respective sizes are $N_{dof} \times 1$ and $N_{dof} \times 1$, with N_{dof} the number of degrees of freedom.

We defined the problem in the MOR framework: we assume the solution space is embedded in a manifold of a reduced dimension $n \ll N_{dof}$. Consequently, inputs and outputs should lie in this a sub-space of dimension n with respect to a given degree of approximation. We also expect the rank of the model $\mathcal{K}$ to be n, even if initially it operates in $\mathbb{R}^{N_{dof} \times N_{dof}}$.

Based on those hypotheses, we introduced a global data-driven learning procedure to directly learn the low-rank model $\mathcal{K}_{lr}$ of the linear discrete problem:

$$\mathcal{K}_{lr}\mathbf{U} = \mathbf{F}$$

where $\mathcal{K}_{lr} \in \mathbb{R}^{N_{dof} \times N_{dof}}$ is the low-rank counterpart of $\mathcal{K}$.

To learn the low-rank model $\mathcal{K}_{lr}$ we consider a train set $\mathcal{T}$ (fig.2.1) of $n_{\mathcal{T}}$ input-output vector couples extracted from a numerical simulation of a physical phenomenon:

$$\mathcal{T} = \{(\mathbf{U}, \mathbf{F})^i \in \mathbb{R}^{N_{dof}} \times \mathbb{R}^{N_{dof}},\ i \in [\![0, n_{\mathcal{T}}]\!]\}$$

Where:

- $i \in [\![0, n_{\mathcal{T}}]\!]$ denotes the i-th time step ans $n_{\mathcal{T}}$ the number of snapshots.
- The vector $\mathbf{U}^i$ is the displacement snapshot at the i-th time step.
- The vector $\mathbf{F}^i$ is the associated force snapshot at the i-th time step.

Those vectors must be understood in their most general sense: $\mathbf{U}^i$ involves generalize displacements that cover displacements and rotations, and $\mathbf{F}^i$ involves the generalize forces that cover forces and moments.

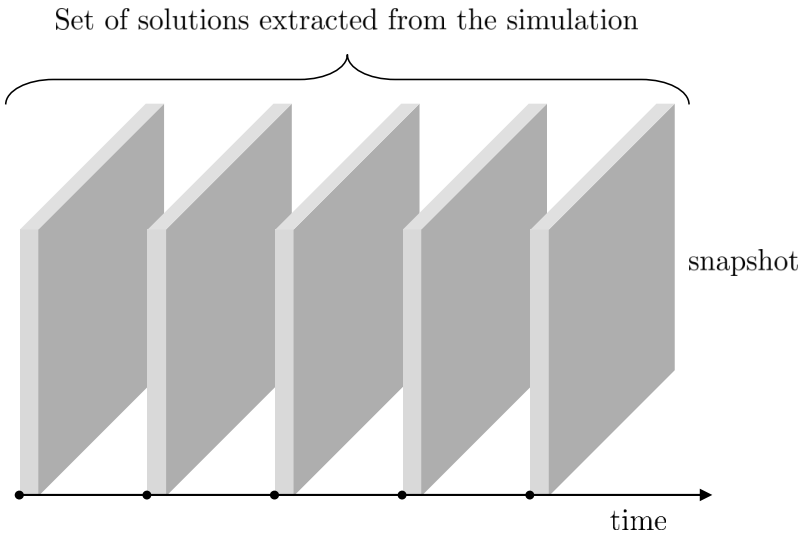


Figure 2.1 – The train set $\mathcal{T}$ contains nodal displacements and forces extracted from a numerical simulation.

We assume the model $\mathcal{K}$ (eq. 2.1) to be expressible from its low-rank form:

$$\mathcal{K} \approx \mathcal{K}_{lr}$$

We introduce a separated representation of $\mathcal{K}_{lr}$ inspired from the separated form proposed in the PGD, the Singular Value Decomposition (SVD, [HH92]) or the CUR decomposition [MD09]:

$$\mathcal{K}_{lr} = \sum_{j=1}^{n} \mathbf{C}_j \otimes \mathbf{R}_j = \sum_{j=1}^{n} \mathbf{C}_j \mathbf{R}_j^{\mathsf{T}} \tag{2.2}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{C}_j$ and $\mathbf{R}_j$ are respectively column and row vectors, and n is the intrinsic dimension of the solution space.

To extract the reduced-order model $\mathcal{K}_{lr}$ from the train set $\mathcal{T}$, we define the following cost functional $\mathbf{E}(\mathcal{K}_{lr})$ whose minimization results in $\mathcal{K}_{lr}$:

$$\mathbf{E}(\mathcal{K}_{lr}) = \sum_{i=1}^{n_{\mathcal{T}}} \left\| \mathbf{F}^i - \mathcal{K}_{lr} \mathbf{U}^i \right\|_F$$

with $\| \bullet \|_F$ the standard Frobenius norm. Then, the previous expression reads in matrix form:

$$\mathbf{E}(\mathcal{K}_{lr}) = \left\| \mathcal{F} - \mathcal{K}_{lr} \mathcal{U} \right\|_F$$

Where $\mathcal{U} = (\mathbf{U}^1, ..., \mathbf{U}^{n_{\mathcal{T}}})$ is the matrix containing in its column the vectors $\mathbf{U}^i$ and $\mathcal{F} = (\mathbf{F}^1, ..., \mathbf{F}^{n_{\mathcal{T}}})$ is the matrix containing in its column the vectors $\mathbf{F}^i$.

This procedure ensures that when addressing the linear problem (eq. 2.1), n linearly independent loading $\mathbf{F}^i, i = 1, ..., n$, are sufficient for the construction of a rank-n model $\mathcal{K}_{lr}$.

2.1.2 Prediction phase

Once the learning phase is completed, the global rank-n model is ready to operate. We consider a new input $\mathbf{U} \notin \mathcal{T}$ sampled from a simulation that was not employed to

construct the train set $\mathcal{T}$. The prediction of the related forces $\hat{\mathbf{F}}$ is simply obtained by a matrix product:

$$\hat{\mathbf{F}} = \mathcal{K}_{lr}\mathbf{U}$$

2.2 Local and incremental learning procedure

Local approximations of the solution space are an appealing strategy when a global reduced-order model increases computational time or results in inaccurate predictions, depending on the size of the reduced-order basis. Thus, we also proposed a local MOR technique to learn a ROM from simulation data. As stated before, a local approach is an efficient procedure to predict the dynamic behavior of spot welds. This section reports the model construction and the generalization to new data.

Inspired from the rank-n constructor, we developed a dictionary of local linear mappings associated with an interpolation function to relate inputs to outputs. We call this learning technique the rank-1 constructor in the iDMD [Rei+19] framework.

In this section, we first present the construction of the dictionary, then we discuss the definition of the interpolation function and finally we illustrate the approach with an example.

2.2.1 Learning a dictionary of rank-1 models

We construct a dictionary $\mathfrak{D}$ of local linear rank-1 models that locally approximate the manifold:

$$\mathfrak{D} = \left\{\mathcal{K}_{lr}^{1}, ..., \mathcal{K}_{lr}^{n_\mathcal{T}}\right\}$$

Each local model $\mathcal{K}_{lr}^{i}$ is related to a couple $(\mathbf{U}, \mathbf{F})^i, i \in [\![0, n_\mathcal{T}]\!]$ extracted from high-fidelity simulations requiring expensive computations. When the train set $\mathcal{T}$ contains several simulations, i simply denotes the sample number and is no longer directly related to the time step.

To construct the dictionary $\mathfrak{D}$, we proceed incrementally. Let us consider the first sampled datum: the pair $(\mathbf{F}^1, \mathbf{U}^1)$. Using the decomposition in the equation 2.2, the first rank-1 model reads:

$$\mathcal{K}_{lr}^{1} = \mathbf{C}_1 \mathbf{R}_1^{\mathrm{T}}, \tag{2.3}$$

The column and row vectors are identified from the minimization of:

$$\mathcal{K}_{lr}^{1} = \operatorname{argmin} \left\| \mathbf{F}^1 - \mathcal{K}_{lr}^{1} \mathbf{U}^1 \right\|_F, \tag{2.4}$$

Using a left multiplication with the transpose vector $\mathbf{F}^{1^{\mathrm{T}}}$ we obtain a symmetric matrix, as it is the case for rigidity matrices, with the benefit of avoiding the ill-conditioned matrix inversion of $(\mathbf{U}^1 \mathbf{U}^{1^{\mathrm{T}}})$:

$$\mathcal{K}_{lr}^{1} = \frac{\mathbf{F}^1 \mathbf{F}^{1^{\mathrm{T}}}}{\mathbf{F}^{1^{\mathrm{T}}} \mathbf{U}^1}.$$

We identify the column vector from the decomposition in the equation 2.3:

$$\mathbf{C}_1 = \mathbf{F}^1 \tag{2.5}$$

and for the row vector:

$$\mathbf{R}_1 = \frac{\mathbf{F}^1}{\mathbf{F}^{1^{\mathrm{T}}} \mathbf{U}^1} \tag{2.6}$$

When the second datum $(\mathbf{F}^2, \mathbf{U}^2)$ arrives we also compute its associated rank-1 approximation $\mathcal{K}_{lr}^{2}$. Then, we proceed progressively with any new datum $(\mathbf{F}^i, \mathbf{U}^i)$ to construct the associated model $\mathcal{K}_{lr}^{i}$.

This local model construction has four main advantages. First, it does not require heavy computations to minimize the equation 2.4. Indeed, the two column and row vectors can be identified using the equation 2.3 and the analytical result of the minimization (eq. 2.4). Second, a simple tensor product is required to build the local model. Hence, a dictionary $\mathfrak{D}'$ filled with couples of tensors of respective sizes $(N_{dof} \times 1)$ and $(1 \times N_{dof})$ can be used, instead of the dictionary $\mathfrak{D}$ of matrices of size $N_{dof} \times N_{dof}$.

$$\mathfrak{D}' = \{(\mathbf{C}^1, \mathbf{R}^1), ..., (\mathbf{C}^{n_\mathcal{T}}, \mathbf{R}^{n_\mathcal{T}})\}$$

Third, these local models have the benefit to combine linearly all the components of $\mathbf{U}^i$ to predict any component of $\mathbf{F}^i$. For instance, a displacement in the direction y in a node can influence the force in the x direction in another node. Finally, as soon a datum is sampled, a model (locally accurate) can be constructed, contrary to the POD for instance where several snapshots must be extracted to build a mapping.

2.2.2 Prediction phase with the dictionary

As soon as the dictionary is filled with models (or at least one model), we can predict the output related any new input $\mathbf{U} \notin \mathcal{T}$.

Each local model $\mathcal{K}_{lr}^i$ in $\mathfrak{D}$ is labeled with the input $\mathbf{U}^i$ used in its construction.

$$\mathfrak{D} = \begin{Bmatrix} \mathcal{K}_{lr}^1, & \cdots & , \mathcal{K}_{lr}^{n_\mathcal{T}} \\ \downarrow & & \downarrow \\ \mathbf{U}^1, & \cdots & , \mathbf{U}^{n_\mathcal{T}} \end{Bmatrix}$$

The model employed for the prediction is denoted $\mathcal{K}(\mathbf{U})$. It is interpolated from the just defined rank-1 models in the dictionary $\mathfrak{D}$ according to:

$$\mathcal{K}_{lr}(\mathbf{U}) \approx \sum_{i=1}^{n_\mathcal{T}} \mathcal{K}_{lr}^i \mathcal{I}(\mathbf{U}^i) \tag{2.7}$$

with $\mathcal{I}(\mathbf{U}^i)$ the interpolation function operating in the input space. This function is assumed decreasing with the distance between $\mathbf{U}$ and $\mathbf{U}^i$. Many functions could be used, in the application on spot welds in chapter 3 we used the nearest neighbor interpolation.

2.2.3 Illustration on the diffusion

We illustrated this procedure on a simple example for explanation purposes. We considered a uniform rod of length l with non-uniform temperature lying on the x-axis from $x = 0$ to $x = l$ heated with a source $\mathbf{S}$. The rod is insulated except on the ends.

The temperature in the rod $[0, l]$ during a simulation time interval $(0, T^f]$ is governed by the following partial differential equation:

$$\frac{\partial \mathbf{u}}{\partial t} - \alpha \frac{\partial \mathbf{u}^2}{\partial x^2} = \mathbf{s} \tag{2.8}$$

with:

- $\mathbf{u}(x, t)$ the temperature in the rod,
- $\mathbf{s}(x, t)$ the heat source in the rod,
- α the thermal diffusivity of the medium.

And the initial and boundary conditions:

- $\mathbf{u}(0, t) = \mathbf{0}$ and $\mathbf{u}(l, t) = \mathbf{0}$
- $\mathbf{u}(0, x) = \mathbf{u}_0$

We illustrated the three main steps of the iDMD: the data generation to construct the training set $\mathcal{T}$, the learning phase on the training set $\mathcal{T}$ and finally the prediction on new inputs.

We used finite differences to approximate the solution and generate the data. Let Δt the time step and $\Delta x = \dfrac{l}{N_{dof} - 1}$ the spatial step with N_{dof} the number of nodes in the interval $[0, l]$.

The fully discretized explicit time-scheme read:

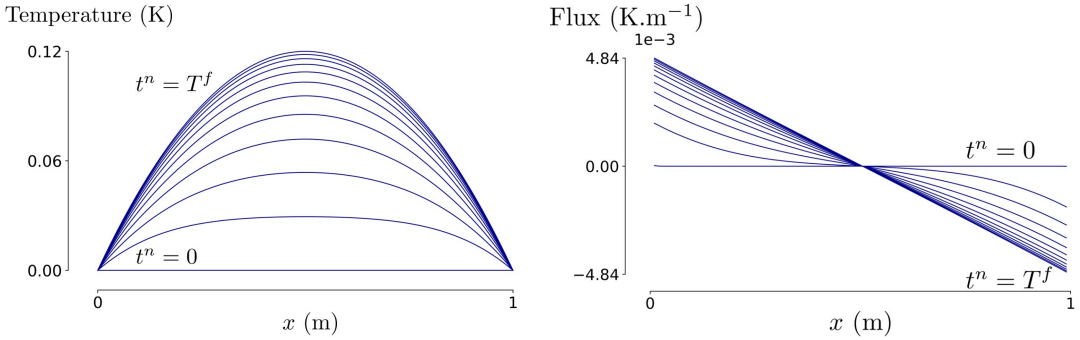
$$\frac{1}{\Delta t}(\mathbf{U}_j^{i+1} - \mathbf{U}_j^i) - \frac{\alpha}{\Delta x^2}(\mathbf{U}_{j+1}^i - 2\mathbf{U}_j^i + \mathbf{U}_{j-1}^i) = \mathbf{S}_j^i \tag{2.9}$$

With:

- i denotes the i-th iteration and $t^i = i\Delta t$,
- $\mathbf{U}_j^i = \mathbf{u}(j\Delta x, i\Delta t),\ j = 1, ..., N_{dof} - 1$ the discretized counterpart of $\mathbf{u}$,
- $\mathbf{S}_j^i = \mathbf{s}(j\Delta x, i\Delta t),\ j = 1, ..., N_{dof} - 1$ the discretized counterpart of $\mathbf{s}$,

We considered as input the temperature in the rod $\mathbf{U}^i = \mathbf{U}(t^i)$ and as output the temperature gradient $\mathbf{F}(t^i)$ denoted $\mathbf{F}^i \in \mathbb{R}^{N_{dof}}, i = 0, ..., n_{\mathcal{T}}$. The temperature gradient is given by $\mathbf{F}^i_j = \frac{1}{\Delta x}(\mathbf{U}^i_j - \mathbf{U}^i_{j-1}), j = 1, ..., N_{dof}$ and $\mathbf{F}^i_1 = \frac{1}{\Delta x}(\mathbf{U}^i_2 - \mathbf{U}^i_1)$.

We constructed the train set $\mathcal{T}$ by extracting $n_{\mathcal{T}} = 12$ snapshots (fig.2.2) from the resolution of equation 2.8 using the integration scheme in equation 2.9. The following set of parameters was employed: $N_{dof} = 100$, $l = 1$, $\Delta t = 3.10^{-5}$, $\mathbf{S}^i = [0, 1, ..., 1, 0]$, and $\alpha = 1$.



(a) Snapshots of training inputs

(b) Snapshots of training outputs

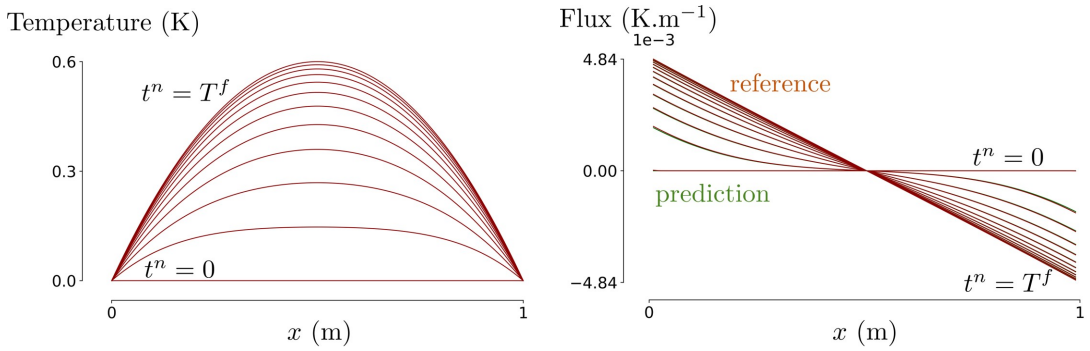
Figure 2.2 – The train set $\mathcal{T}$ of the thermal illustration

We then proceeded to the learning phase: the dictionary construction and the interpolation function definition. We computed a rank-1 model from each couple $(\mathbf{U}^i, \mathbf{F}^i)$ and stored it in the dictionary $\mathfrak{D}$ using the decomposition from equation 2.2:

$$\mathfrak{D} = \{(\mathbf{C}^1, \mathbf{R}^1), ..., (\mathbf{C}^{n_{\mathcal{T}}}, \mathbf{R}^{n_{\mathcal{T}}})\}$$

We defined the interpolation function as the nearest neighbor interpolation, *i. e.* the local model chosen is related to the closest temperature vector with respect to the L_2 norm.

Finally, we proceeded to the prediction phase. We considered new temperature inputs (fig.2.3). The temperature and gradient snapshots were extracted from a simulation using a different heat source $\mathbf{S}^i = [0, 5, ..., 5, 0]$. We then recovered the flux $\hat{\mathbf{F}}$ associated with a new temperature input $\mathbf{U}$ with the local approach (fig.2.3) using equation 2.7.



(a) Snapshots of test inputs

(b) Snapshots of test outputs

Figure 2.3 – Promising results were obtained as reference and predicted outputs are barely distinguishable.

We obtained excellent results on this very simple illustration. To move towards an industrial use, we address in the next section some high dimensional issues engineers face in crash test simulation during a car project development.

2.3 Dealing with high-dimensional spaces

Several high-dimensional problems are encountered in crash test simulations during the development of a vehicle project. Indeed, these simulations involve many degrees of freedom $N_{dof} \gg 1$ and dynamic behaviors that depend on the mechanical history. Moreover, they are used in a multi-query design framework. These high-dimensional problems are related to the "curse of dimensionality" [AK18]. In this section, we present approaches to address these issues with the iDMD.

To tackle these concerns, we extended the parametric space of the iDMD using new parameters. Those parameters account for design queries and mechanical history. We also associated the iDMD with reduced basis to operate on high-dimensional spaces ($N_{dof} \gg 1$).

In this section, we focus on the consideration of the mechanical history, then discuss the learning parametrization in a multi-query design framework, and finally introduce the iDMD associated with reduced bases.

2.3.1 Considering the mechanical history

During a crash simulation, the spot welds experience irreversible deformations such as plasticity, damage and fracture. Those deformations depend on the time evolution, also called history, of the loading. However, the global and local approaches presented previously are processes without time-memory, they do not consider the evolution of the system. We propose to consider the loading history with the inputs $\mathbf{U}^i$. Additionally, a parametrization of those inputs can be necessary to consider irreversible deformations, for instance the cumulative work.

We will see later in chapter 3 that the loading in the use case on spot welds was monotonic and therefore so were the displacements. Therefore, using those displacements sufficed to consider the history of loading.

However, if the experiment had involved a loading phase followed by an unloading phase (fig. 2.4), the displacements would not account anymore for the history. In this case, we could have used the cumulative plastic work or the cumulative plastic deformation.

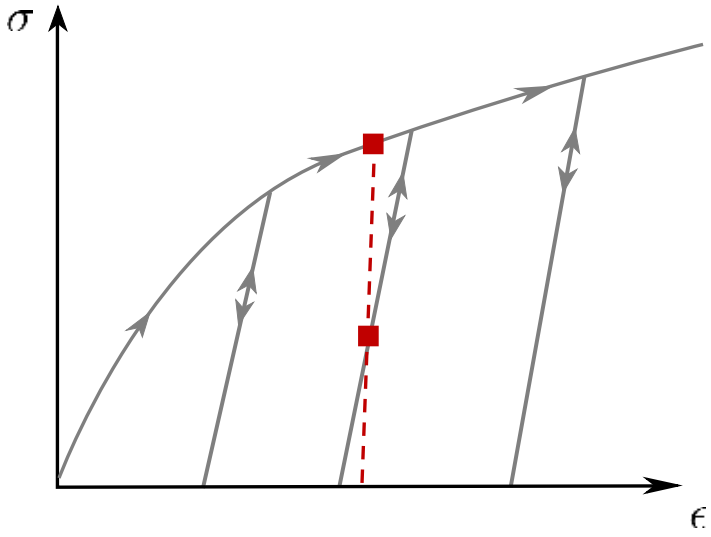


Figure 2.4 – When loading and then unloading, the same deformation (and hence displacement) is related to several stress states in a plastic behavior.

2.3.2 Learning parametric models

We propose to extend the iDMD parametric setting to deal with design parameters as the car design process (introduced in chapter 1) is a multi-query design process. This design process impacts the modeling of the spot weld as engineers test multiple alloys in the assembly and various plate thicknesses to converge on the optimal balance between weight and structural strength. Inspired from the PGD rationale, we propose to associate the reduced models to inputs (the displacements) and design parameters grouped into the vector $\boldsymbol{\mu}$= ($\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_{\mathtt{P}}$).

As soon as the models $\mathcal{K}(\boldsymbol{\mu}_p)$, $p = 1, \ldots, \mathtt{P}$, are constructed, the one related to $\boldsymbol{\mu}$ is computed from:

$$\mathcal{K}(\boldsymbol{\mu}) \approx \sum_{p=1}^{\mathtt{P}} \mathcal{K}(\boldsymbol{\mu}_p)\mathcal{J}_p(\boldsymbol{\mu}),$$

with $\mathcal{J}_p(\boldsymbol{\mu})$ the interpolation functions operating in the parametric space, assumed decreasing with the distance between $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}$.

2.3.3 Learning procedure in extremely high dimensional spaces

When the problem considered involves numerous degrees of freedom $N_{dof} \gg 1$, the usual functions that we could consider for the interpolation $\mathcal{I}$ of local models (eq. 2.7) in the prediction phase fail. Indeed, the neighborhood density tend to decrease in high dimensional spaces (fig. 2.5).

To alleviate the interpolation issues, we defined a reduced counterpart of inputs and outputs. We suggest retrieving first an $r-$dimensional reduced basis from the vectors in $\mathcal{U}$ and group the reduced basis vectors in the matrix $\mathcal{B} \in \mathbf{R}^{N_{dof} \times r}$. Thus, the displacement vectors can be expressed as:

$$\mathbf{U}^i = \mathcal{B}\mathbf{v}^i$$

where $\mathbf{v}^i$ is the displacement projected on the reduced basis.

Therefore we can express the discrete system (2.1) in the reduced form:

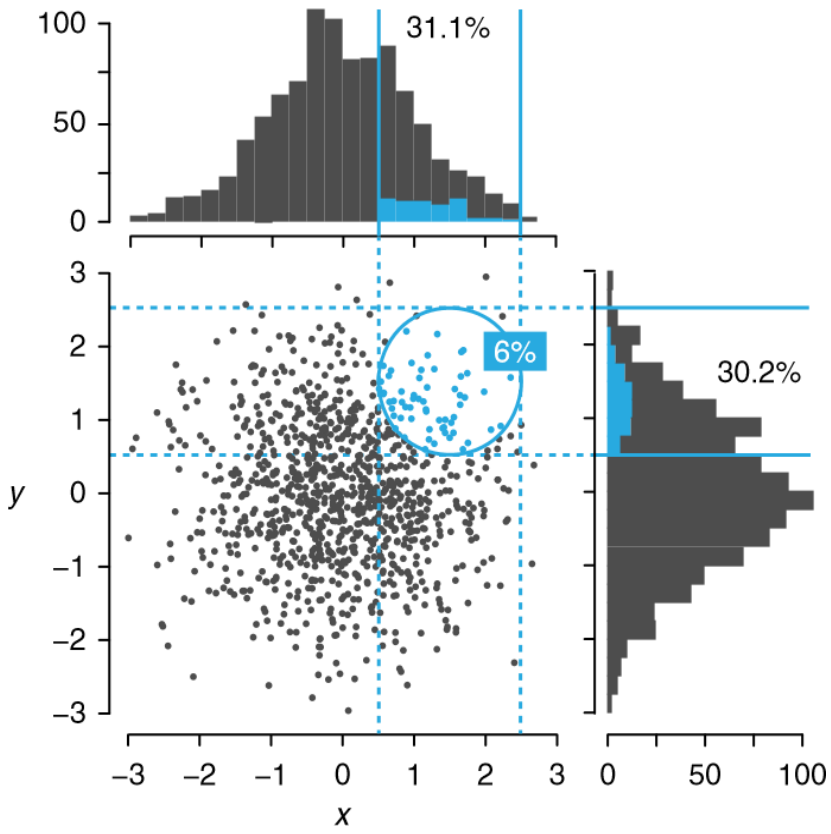


Figure 2.5 – As the dimensionality increases, the "volume" that the samples may occupy grows rapidly. Any given neighborhood of the space is more likely to contain no data and thus be sparse [AK18].

$$\left(\mathcal{B}^T \mathcal{K}_{lr} \mathcal{B}\right) \mathbf{v}^i = \mathcal{B}^T \mathbf{F}^i$$

or in its more compact counterpart:

$$\mathcal{K}_{lr} \mathbf{v}^i = \mathbf{f}^i.$$

In the prediction phase, we compute first the reduced forms of new inputs, *i.e.* inputs that do not belong to $\mathcal{U}$ and $\mathcal{F}$:

$$\mathbf{v}^i = \mathcal{B}^T \mathbf{U}^i$$

and,

$$\mathbf{f}^i = \mathcal{B}^T \mathbf{F}^i$$

Then, the reduced model $\mathcal{K}_{lr}$ is constructed from the reduced couples $(\mathbf{v}^i, \mathbf{f}^i)$ using the rank-n or the rank-1 approaches introduced in the iDMD.

Usually when constructing the train set, the data is filtered to reduce noise after sampling. This filtration and the projection using the matrix $\mathcal{B}$ are similar processes as high-frequency modes are discarded in the reduced basis construction.

This procedure is appropriate to circumvent interpolation issues in high-dimensional spaces, however, it is less physically sound than the rank-n and the rank-1 procedure. Indeed, inputs and outputs lose their physical significance as soon as they are reduced.

2.4 Revisiting mechanical condensation

We proposed in chapter 1 to investigate the integration of spot weld ROMs instead of mesoscale models in crash test solvers to compute high-fidelity simulations in less than a day. Thus, we first introduced the iDMD to learn the ROM of a spot weld. This section reports the integration of the ROM in the global model. We referred to the mechanical condensation introduced in chapter 1 for the integration procedure and studied the condensation on a static and dynamic cases to develop the local-global approach.

We found the validity of a condensed model depends on the existence of external forces acting inside the region concerned by the condensation procedure and the consideration of the previous states for time-dependent models. By formulating hypotheses based on these conditions, we can perform the coupling of the local model, where the condensation applies, to the global one by learning a ROM of the interface separating

the local domain from the global one (fig. 2.6). This ROM can infer forces at the nodes on the boundary between the two domains, for any displacement applied on the global domain.

In this section, we report first the conditions for the condensation in a static case, and then we focus on a dynamical case. Finally, we present the local-global integration framework.

2.4.1 Condensed static models

We investigated first the mechanical condensation in a static case. We considered the domain Ω distinguished in two sub-domains $\Omega = \tilde{\Omega} \cup \omega$ (fig. 2.6). We called "global domain" the sub-domain $\tilde{\Omega}$ which requires a coarse description, whereas the other sub-domain ω, called "local domain" requires a fine description which necessarily has an impact on CPU time. The local domain ω is represented with a 3D fine discretization surrounded by a transition zone discretized in 2D (*e.g.* using shell elements) to connect with the global domain where a discrete 2D model applies.

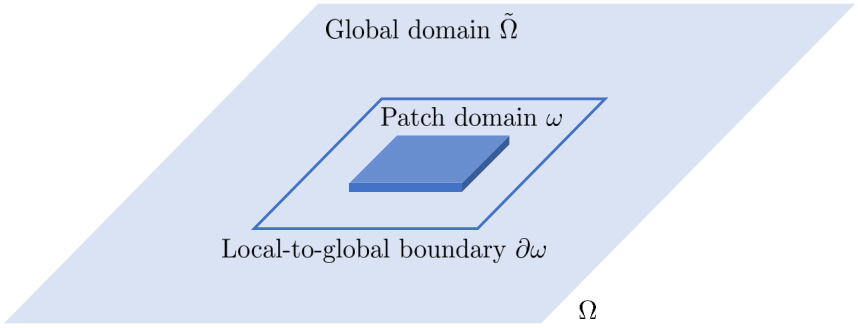


Figure 2.6 – Local domain ω connected to a global domain $\tilde{\Omega}$ through the local-to-global boundary $\partial\omega$

The domain Ω is governed by an elasto-statics equation which discrete form reads:

$$\mathcal{K}\mathbf{U} = \mathbf{F},$$

where $\mathbf{U}$ represents the vector that contains the nodal displacements and $\mathbf{F}$ stands for

the nodal external loads. The vectors $\mathbf{U}$ and $\mathbf{F}$ can be decomposed in $\mathbf{U} = (\mathbf{U}_g, \mathbf{U}_b, \mathbf{U}_i)$ and $\mathbf{F} = (\mathbf{F}_g, \mathbf{F}_b, \mathbf{F}_i)$ with:

- $\mathbf{U}_g$ and $\mathbf{F}_g$ the displacements and forces in the nodes of the global domain $\tilde{\Omega}$,
- $\mathbf{U}_b$ and $\mathbf{F}_b$ the displacements and forces in the nodes of the local-to-global boundary $\partial\omega$,
- $\mathbf{U}_i$ and $\mathbf{F}_i$ the displacements and forces in the nodes of the local domain ω.

We focus on the nodes on local-to-global boundary $\mathbf{U}_b$ and on the internal nodes $\mathbf{U}_i$. The problem can be decomposed into:

$$\begin{pmatrix} \mathcal{K}_{bb} & \mathcal{K}_{bi} \\ \mathcal{K}_{ib} & \mathcal{K}_{ii} \end{pmatrix} \begin{pmatrix} \mathbf{U}_b \\ \mathbf{U}_i \end{pmatrix} = \begin{pmatrix} \mathbf{F}_b \\ \mathbf{F}_i \end{pmatrix}. \tag{2.10}$$

To reduce the number of degrees of freedom N_{dof}, we sook for an application condensing the internal degrees of freedom on the boundary:

$$\begin{pmatrix} \mathbf{U}_b \\ \mathbf{U}_i \end{pmatrix} = \mathcal{P}\mathbf{U}_b \tag{2.11}$$

To condense the internal degrees of freedom, we developed the second relationship in (eq. 2.10) and expressed the internal nodes $\mathbf{U}_i$ using the other variables:

$$\mathcal{K}_{ib}\mathbf{U}_b + \mathcal{K}_{ii}\mathbf{U}_i = \mathbf{F}_i$$

Thus,

$$\mathbf{U}_i = \mathcal{K}_i^{-1}\mathbf{F}_i - \mathcal{K}_{ii}^{-1}\mathcal{K}_{ib}\mathbf{U}_b$$

Consequently, the first relationship in the equation 2.10 results in

$$(\mathcal{K}_{bb} - \mathcal{K}_{bi}\mathcal{K}_{ii}^{-1}\mathcal{K}_{ib})\mathbf{U}_b = \mathbf{F}_b - \mathcal{K}_{bi}\mathcal{K}_{ii}^{-1}\mathbf{F}_i,$$

which can be more compactly rewritten as:

$$\tilde{\mathcal{K}}_{bb}\mathbf{U}_b = \mathbf{F}_b - \tilde{\mathbf{F}}_i, \tag{2.12}$$

with $\tilde{\mathbf{F}}_i = \mathcal{K}_{bi}\mathcal{K}_{ii}^{-1}\mathbf{F}_i$ and $\tilde{\mathcal{K}}_{bb} = \mathcal{K}_{bb} - \mathcal{K}_{bi}\mathcal{K}_{ii}^{-1}\mathcal{K}_{ib}$

We concluded from equation 2.12 that the boundary displacements depend on boundary forces and external forces applied on the nodes in the local domain ω. If there are no external forces on the internal nodes, i.e $\mathbf{F}_i = \mathbf{0}$, then $\tilde{\mathbf{F}}_i = \mathbf{0}$ and consequently the condensation simply involves $\mathbf{U}_b$ and $\mathbf{F}_b$. Therefore the boundary displacements only depend on the boundary forces $\mathbf{F}_b$.

The matrix $\tilde{\mathcal{K}}_{bb}$ can be easily learnt from data in the 1D case using appropriate displacements conditions. Indeed, $\mathbf{U}_b$ and $\mathbf{F}_b$ contain in this configuration two components that represent the two boundary nodes as they are defined in $\mathbb{R}^2$. If we apply $\mathbf{U}^\intercal = (1, 0)$ and solve the associated static mechanical linear problem, the resulting $\mathbf{F}_b$ will represent the first column of $\tilde{\mathcal{K}}_{bb}$. The solution $\mathbf{F}_b$ related to $\mathbf{U}^\intercal = (0, 1)$ will then give the second column of $\tilde{\mathcal{K}}_{bb}$;

If there are external forces applied on ω, then $\mathbf{F}_i \neq \mathbf{0}$ and the problem involves internal variables represented by the components of $\tilde{\mathbf{F}}_i$. The (maximum) number of internal variables is the size of $\tilde{\mathbf{F}}_i$ independently of the intrinsic richness of the loading $\mathbf{F}_i$. For instance in the 1D case, there are only two internal variables, the two components of $\tilde{\mathbf{F}}_i$. Nevertheless, computing these two internal variables needs an extra-calculation, for example if $\mathbf{U}_b = \mathbf{0}$, then $\mathbf{F}_b = \tilde{\mathbf{F}}_i$.

2.4.2 Condensed dynamic models

We studied the effects of time derivatives on the domain decomposition and condensation when the domain Ω experienced dynamical behaviors following two alternative approaches. The first approach operates from the discretized form of the system of equation (eq. 2.13) and the second from the static condensation. The former is illustrated on a linear model with first order derivatives, while the second is illustrated on the structural dynamics setting introduced in chapter 1, which involves second order time derivatives (eq. 2.14).

We considered the generic semi-discretized linear first order dynamical system in Ω:

$$\mathcal{C}\dot{\mathbf{U}}(t) + \mathcal{D}\mathbf{U}(t) = \mathbf{F}(t) \tag{2.13}$$

where $\mathcal{C}$ is the damping matrix and $\mathcal{D}$ the stiffness matrix. The vector $\mathbf{U}$ contains the nodal displacements and $\dot{\mathbf{U}}$ the associated velocities while the vector $\mathbf{F}$ stands for the nodal external loads.

The time-discretization is performed with Euler's implicit time scheme. The fully discretized system reads:

$$\mathcal{C}'\mathbf{U}^n + \mathcal{D}\mathbf{U}^n = \mathbf{F}^n + \mathcal{C}'\mathbf{U}^{n-1}$$

with Δt the time step and $\mathcal{C}' = \mathcal{C}/\Delta t$.

We introduced the matrix $\mathcal{K} = \mathcal{C}' + \mathcal{D}$ and $\mathbf{G} = \mathcal{C}'\mathbf{U}^{n-1}$ for a more compact equation. Therefore, the previous equation results in:

$$\mathcal{K}\mathbf{U} = \mathbf{F}^n + \mathbf{G}^n$$

As in the static case (eq. 2.10), we decomposed the domain and focused on the local-to-global boundary nodes, with $\mathbf{U}_b$ and $\mathbf{F}_b$ representing the boundary forces and displacements, and on the internal nodes with $\mathbf{U}_i$ and $\mathbf{F}_i$ representing the internal displacements and forces. The objective was to express the boundary displacements from the other variables. As the matrix $\mathcal{G}_b$ vanishes on the domain boundary where boundary conditions apply, the rationale employed in the previous section leads to:

$$\tilde{\mathcal{K}}_{bb}\mathbf{U}_b = \mathbf{F}_b - \tilde{\mathbf{F}}_i + \mathbf{G}_i$$

In the 1D case, the two internal variables in the vector $\tilde{\mathbf{F}}_i + \mathbf{G}_i$ are different at each time step. Indeed, $\mathbf{G}_i$ depends on $\mathbf{U}^{n-1}$. Consequently, the dependence of the condensed model on the solution at the previous time step seems difficult to evidence and practically implement.

We consider now the second approach in which we revisit static condensation. This

procedure is illustrated on a semi-discretized elastodynamic problem:

$$\mathcal{M}\ddot{\mathbf{U}} + \mathcal{C}\dot{\mathbf{U}} + \mathcal{K}\mathbf{U} = \mathbf{F} \tag{2.14}$$

where $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$ are respectively the mass, damping and stiffness matrices. The vector $\mathbf{U}$ contains the nodal displacements. The vectors $\ddot{\mathbf{U}}$ and $\dot{\mathbf{U}}$ are the associated accelerations and velocities. The vector $\mathbf{F}$ stands for the nodal external loads.

We took inspiration from the relation (eq. 2.10) which neglects inertia. Thus, a mapping relates the internal degrees of freedom with the ones on the boundary as in the static case:

$$\mathbf{U}_i = \mathcal{K}_{ii}^{-1}\mathbf{F}_i - \mathcal{K}_{ii}^{-1}\mathcal{K}_{ib}\mathbf{U}_b,$$

We considered no external forces were applied on the internal nodes. Therefore, we assumed from the rational in the static case (eq. 2.10) that the nodal displacements on the internal and boundary nodes depend on the displacements on the boundary:

$$\begin{pmatrix} \mathbf{U}_b \\ \mathbf{U}_i \end{pmatrix} = \begin{pmatrix} \mathcal{I} \\ -\mathcal{K}_{ii}^{-1}\mathcal{K}_{ib} \end{pmatrix} \mathbf{U}_b = \mathcal{P}\mathbf{U}_b. \tag{2.15}$$

We expressed the dynamic problem using the assumption in (eq. 2.15):

$$\mathcal{P}^{\mathrm{T}}\mathcal{M}\mathcal{P}\ddot{\mathbf{U}}_b + \mathcal{P}^{\mathrm{T}}\mathcal{C}\mathcal{P}\dot{\mathbf{U}}_b + \mathcal{P}^{\mathrm{T}}\mathcal{K}\mathcal{P}\mathbf{U}_b = \mathcal{P}^{\mathrm{T}}\mathbf{F},$$

This approach does not depend on the solution in internal nodes at the previous time step contrary to the previous procedure. However, its validity and accuracy strongly depend on the assumption (eq. 2.15), that works well when the inertia is moderate. Hence, static condensation seems a reasonable approach for the local-global integration of condensed spot weld models, except maybe close to the rupture stage.

2.4.3 Local-global integration scheme

We present now the time integration of the condensed ROM in a global solver. We consider the domain Ω experiencing inelastic deformations as previously introduced. This domain is distinguished into the local domain ω and the global domain $\Omega = \tilde{\Omega} \cup \omega$

(fig. 2.7). We construct the condensed ROM of the local domain using the iDMD and the condensed static approach for dynamic models. We summarized this approach in the figure 2.8. By integration a condensed ROM in the global domain the simulation time decreases because we do not account anymore for:

- the tiny time step usual explicit simulation imply,
- the extremely fine meshes required for adequately describing the local domain.

We assume the following hypothesis from the discussion in the section 2.4.2 are verified to implement the condensed ROM in $\boldsymbol{\Omega}$:

- The external forces do not apply inside the patch ω,
- The dynamical effects (inertia) in the patch domain are negligible compared to the mechanical effects.

(a) global domain $\tilde{\Omega}$ (b) local domain ω

Figure 2.7 – Domain Ω distinguished in two sub-domains.

The reduced-order model construction within the iDMD framework starts with a learning phase as the two approaches introduced are data-driven. We apply a variety of loading on the boundary $\partial\Omega$ of the domain Ω. Thus, the local domain ω experiences a wide range of mechanical deformations up until rupture.

We extract $n_{\mathcal{T}}$ couples of displacements and forces from the resolution of several inelastic problems in Ω. The train set $\mathcal{T}$ contains nodal displacement and forces extracted on the local-to-global boundary $\partial\omega$:

$$\mathcal{T} = \{(\mathbf{U}_b, \mathbf{F}_b)^i \in \mathbb{R}^{N_{dof}} \times \mathbb{R}^{N_{dof}},\ i \in [\![0, n_{\mathcal{T}}]\!]\}$$

where N_{dof} scales with the number of nodes on $\partial\omega$ multiplied by the number of degrees of freedom per node and $n_{\mathcal{T}}$ is the number of snapshots in $\mathcal{T}$.

The local-to-global boundary stiffness $\mathcal{K}_{lr}$ is learnt on the train set $\mathcal{T}$ according to the rank-n or rank-1 approaches presented in the sections 2.1 and 2.2. This learning denotes the final step in the offline stage.

We then integrate the condensed ROM in the global solver following the algorithm 2. At each time step t^n the forces on the nodes on the local-to-global boundary are added in to momentum equation (see line 4). As the local-to-global boundary lies in the area discretized with 2D elements, there are no continuity issues between the 3D and the 2D area to deal with.

Algorithm 2: Global-local time scheme integration

1 **for** $n \leq n_f$ **do**
2 Extract the nodal displacements $\mathbf{U}_b^n$ on the local-to-global boundary $\partial\omega$ from the nodal displacements in $\tilde{\mathbf{\Omega}}$;
3 Evaluate the forces on the boundary $\partial\omega$ using the iDMD;
4 Compute the nodal accelerations in the domain $\tilde{\Omega}$ by solving the momentum balance equation: $\tilde{\mathcal{M}}\ddot{\mathbf{U}}^n = \tilde{\mathbf{F}}_{int}^n + \tilde{\mathbf{F}}_{ext}^n + \mathbf{F}_b^n$ where $\tilde{\bullet}$ refers to quantities defined in $\tilde{\Omega}$ and $\mathbf{F}_b^i$ traduces the local domain effects on $\partial\omega$;
5 Integrate acceleration to compute velocities and displacements at time $t + \Delta t$;
6 **end**

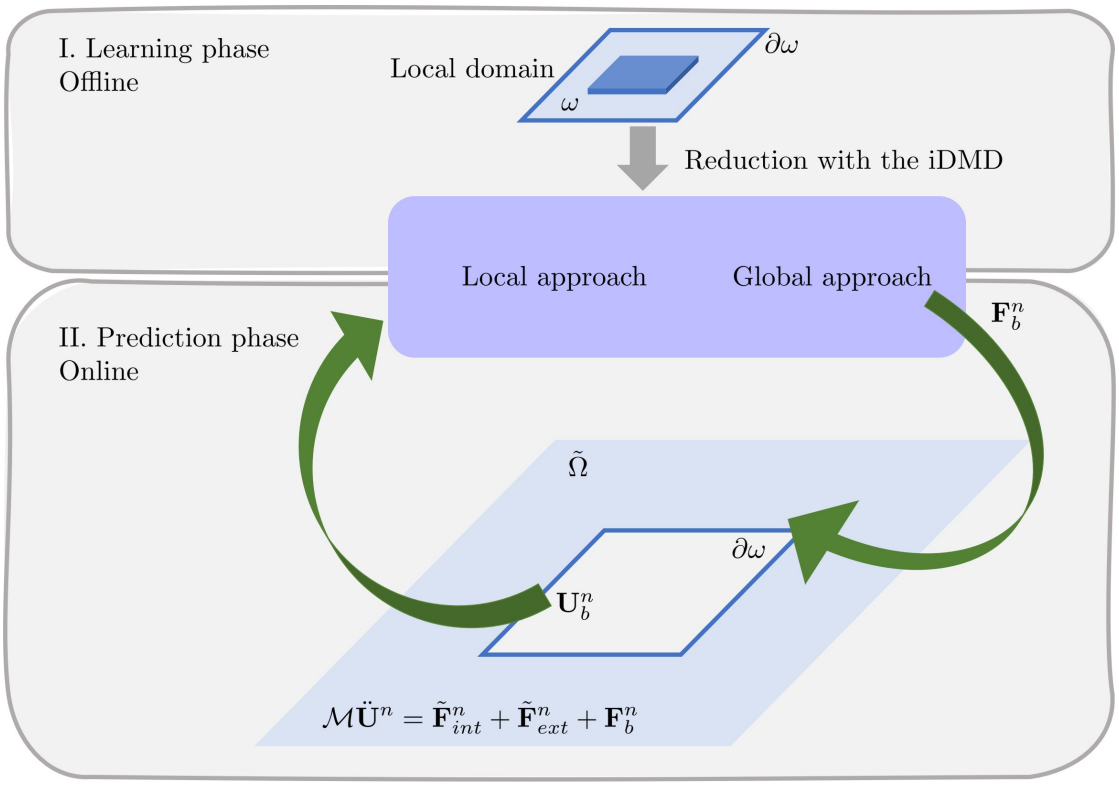


Figure 2.8 – Time integrator making use of the condensed ROM

Chapter 3

Application: spot welds in car crash

Summary

Car manufacturers need fast and accurate solvers to accomplish only design confirmation crash tests. However, the explicit time-scheme integration employed to simulate fast dynamics in the car structure limits the integration of accurate local models, such as mesoscale spot weld models. Therefore, MOR of these models is an appealing strategy to increase crash test simulations' accuracy and reduce computational time. Using the rank-1 approach and the local-global procedure introduced in chapter 2, we investigated whether a reduced-order model of a spot weld and the associated integration framework could improve the crash solver performance. In this chapter, we report the accuracy and the computational time of a spot weld ROM joining two sheets and study the impacts of the local-global integration in VPS crash solver.

For 6 high-fidelity validation simulations, the rank-1 approach reduced the computational time on 16 processors from 1h30 to less than 8 min. The relative error of prediction was lower than 10% before rupture and remained globally acceptable during crack propagation. Additionally, the local-global integration framework demonstrated promising results for the crash solver. Consequently, we started a more extensive study to address complex loading.

In the first section we present the numerical setup and the construction of the training data set. In the second section we focus on the validation of the hypotheses regarding the rank-1 procedure. Finally, we report the rank-1 and local-global performances in the last section.

3.1 Data-set construction

We investigated the reduction of a high-fidelity spot weld model to speed up VPS crash solver when it accounts for accurate spot weld behaviors. To construct this ROM, we chose the iDMD introduced in chapter 2. This procedure is an "a posteriori" MOR technique and learns a ROM from a train set $\mathcal{T}$. Therefore, we defined a numerical experiment to extract training data from high-fidelity simulations.

Using a DoE of five binary parameters, we obtained 28 high-fidelity training simulations of spot welds representing unitary solicitations with VPS. Following the procedure in section 2.4, we extracted snapshots of displacements and forces on interface nodes, the local-to-global boundary nodes, to fill the train set $\mathcal{T}$. Thus, the ROM would learn the behaviors of the local-to-global boundary nodes related to the parametric loading space defined by the DoE.

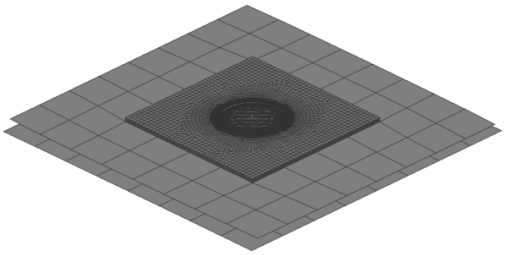
This section introduces first the local domain, then presents the construction of the train set and the data pre-processing.

3.1.1 Local domain

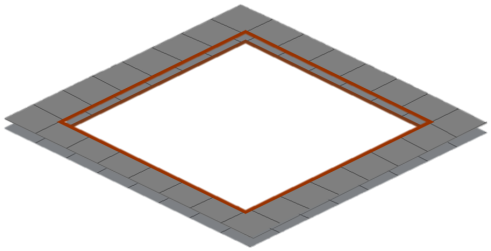
We considered in this use case a spot weld joining two sheets made in $22\mathrm{M_nB_5}$, an alloy that engineers use to lighten structural parts, such as the middle pillar. This part is the structural support on each side of the car body between front and back seats. A car holds on average 80 spot welds in this alloy. Those spot welds are critical during a crash, so the car design attempts to limit their rupture. Therefore, a crash solver that simulates their precise dynamical behaviors would be a valuable tool.

We used VPS to run the high-fidelity simulations needed to build the train set $\mathcal{T}$. This detailed spot weld model is built with the Von-Mises plasticity model, an isotropic hardening model and a rupture model developed by ESI. The spot weld HAZ and the FZ were discretized with numerous finite elements of volume using VPS mesoscale model, while the sheets were represented with shell elements (fig. 3.1a). This model predicts accurate dynamical behaviors of spot welds as it considers the phase's distribution and their mechanical properties, for instance, the plastic strain state after welding in the HAZ.

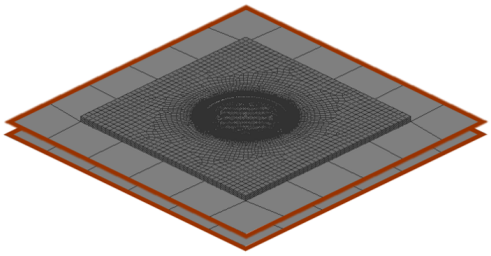
To learn a condensed ROM following the local-global approach (sec. 2.4), we distinguished the domain considered in this numerical setup into two sub-domains $\Omega = \tilde{\Omega} \cup \omega$ (fig. 3.1a). We referred to ω as the local domain and to $\tilde{\Omega}$ as the global domain. The domain ω involved the elements of volume representing the FZ and the HAZ surrounded by a transition region. We equipped this region with a 2D coarser discretization (fig. 3.1c). The global domain $\tilde{\Omega}$ represented the sheets, discretized with shell elements (fig. 3.1b). We defined the local-to-global boundary $\partial\omega$ on the interface connecting the local domain $\partial\omega$ to the global one $\tilde{\Omega}$ (fig. 3.1d). This boundary $\partial\omega$ comprised 48 nodes; 24 nodes were on the interface between the upper sheet and the spot weld, while the remaining 24 nodes were on the interface between the lower sheet and the spot weld. Setting the boundary nodes on shell elements reduced the number of degrees of freedom involved in the reduction (fig. 3.1c). Moreover, as the boundary connects shell elements in the local domain to shell elements in the global domain the integration of the ROM of the boundary is straightforward. This integration is presented later in section 3.3.3.



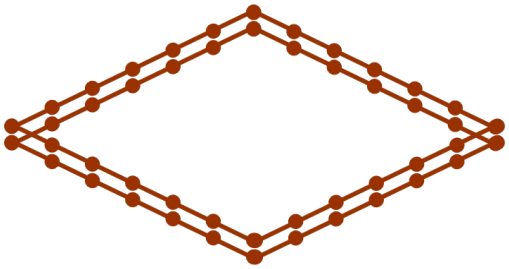
(a) Domain Ω considered in our setup



(b) Global domain $\tilde{\Omega}$



(c) Local domain ω

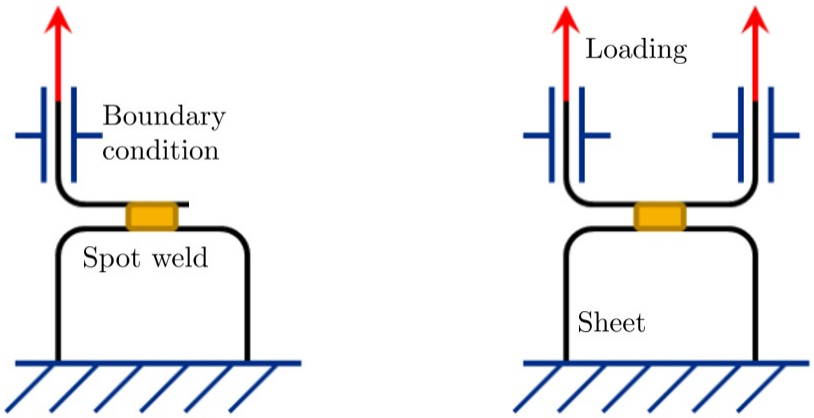


(d) Local-to-global boundary $\partial\omega$

Figure 3.1 – To reduce CPU time, the ROM learns the behaviors of the boundary $\partial\omega$.

3.1.2 Train set construction

To construct our train set $\mathcal{T}$, we assumed in a first approximation that the behaviors of the spot welds during a crash test could be decomposed into unitary behaviors encountered during mechanical tests, such as peeling (fig. 3.2b) and tensile tests (fig. 3.2a).



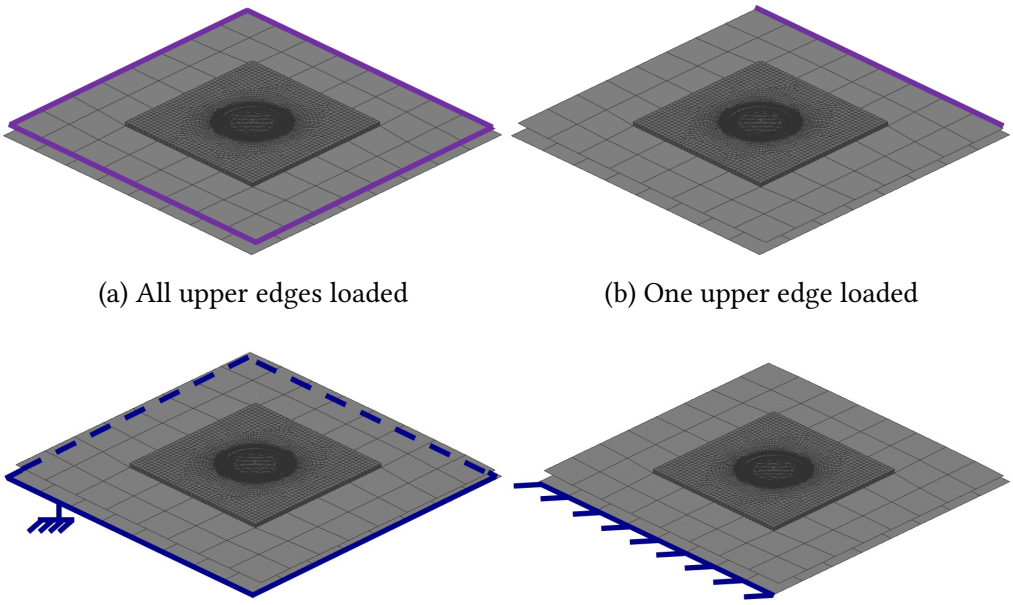
(a) Peeling loading from [Dur+15] (b) Tensile loading from [Dur+15]

Figure 3.2 – Unitary loading conditions

Thus, we defined 5 parameters that drove the loading and degrees of freedom on the boundary $\partial\Omega$ of the domain Ω. The simulations represented behaviors as those observed during the mechanical tests (fig. 3.3). Those parameters involved:

1. the frozen nodes of the lower sheet: all edges (fig. 3.3a) or one edge (fig. 3.3b),
2. the loaded nodes of the upper sheet: all edges (fig.3.3c) or one edge (fig. 3.3d),
3. the loading velocity in the x direction: 0 m.s^{-1} or 0.1 m.s^{-1},
4. the loading velocity in the y direction: 0 m.s^{-1} or 0.1 m.s^{-1},
5. the loading velocity in the z direction: 0 m.s^{-1} or 0.1 m.s^{-1}.

This parameterization of the loading space verified the two assumptions in the definition of the local-global approach (sect. 2.4): first, no external forces applied inside the



(a) All upper edges loaded

(b) One upper edge loaded

(c) All lower edges frozen

(d) One lower edge frozen

Figure 3.3 – The loading was enforced using a velocity condition on the boundary nodes of the domain Ω.

local domain ω, and second, dynamical effects were negligible. Indeed, the loading applied on the boundary $\partial\Omega$ so all the forces the spot weld encountered passed through the boundary $\partial\omega$. Additionally, we used a smooth velocity function for the spot weld to experience quasi-static behaviors. Thus, we assumed the inertia to be negligible.

After the loading parameters were defined, we constructed a complete DoE from the loading parameters to sample the loading space. This DoE contained all the combinatorial combinations of the 5 binary parameters. We excluded the trivial cases where null velocities applied on the boundary $\partial\Omega$, which reduced the DoE size to $2^5 - 2^2 = 28$ different loading cases (tab. 3.1).

Then, we simulated the problems resulting from the loading cases in the DoE and extracted nodal displacements and forces $(\mathbf{U}_b^i, \mathbf{F}_b^i)$ on the boundary $\partial\omega$ to construct the train set $\mathcal{T}$:

$$\mathcal{T} = \{(\mathbf{U}_b, \mathbf{F}_b)^i \in \mathbb{R}^{N_{dof}} \times \mathbb{R}^{N_{dof}},\ i \in [\![0, n_{\mathcal{T}}]\!]\}$$

With:

- $n_{\mathcal{T}} = 1.4 \times 10^6$ couples of snapshots,
- $N_{dof} = 144$ the number of degrees of freedom considered on the boundary $\partial\omega$.

Each high-fidelity computation lasted 1h30 on 16 processors. It simulated 14ms of the spot weld behaviors from the beginning of the loading until after rupture. We extracted 50000 snapshots $(\mathbf{U}_b^i, \mathbf{F}_b^i)$ respectively of size $(N_{dof} \times 1)$ and $(N_{dof} \times 1)$ from each simulation at a sampling period of 3×10^{-4}ms. The loading conditions resulted in null moments on the boundary $\partial\omega$. Thus, the number of degrees of freedom is $N_{dof} = 3 \times 48 = 144$.

The DoE had symmetric loading cases that we could not use to reduce the number of simulations to run. For instance, a 90° clockwise rotation along z transformed the parametrization of the 5th simulation into the parametrization of the 13th simulation. However, we could not exploit those symmetries because when we applied this rotational mapping on the displacements and forces snapshots of the training simulation (5), the resulting forces and displacements did not match the inputs and outputs extracted from the training simulation (13).

We pre-processed the data in the train set $\mathcal{T}$ to reduce noise coming from the dynamical effects and the numerical errors in the high-fidelity simulations. The couples $(\mathbf{U}_b^i, \mathbf{F}_b^i)$ were smoothed with the exponential moving average. Let U_s the smoothed displacements. There are computed according to :

$$U_s^i = \begin{cases} U_b^0, & i = 0 \\ \alpha U_b^i + (1-\alpha) U_s^{i-1}, & i > 0 \end{cases}$$

With α the degree of weighting decrease. This smoothing can be performed recursively online for new inputs and outputs in the prediction phase. The train set $\mathcal{T}$ construction terminated with this final step.

Training simulation	Frozen edges	Loaded edges	Loading velocity (m/s) V_x	V_y	V_z
1	all	all	0	0	0.1
2	all	one	0	0	0.1
3	one	all	0	0	0.1
4	one	one	0	0	0.1
5	all	all	0	0.1	0
6	all	one	0	0.1	0
7	one	all	0	0.1	0
8	one	one	0	0.1	0
9	all	all	0	0.1	0.1
10	all	one	0	0.1	0.1
11	one	all	0	0.1	0.1
12	one	one	0	0.1	0.1
13	all	all	0.1	0	0
14	all	one	0.1	0	0
15	one	all	0.1	0	0
16	one	one	0.1	0	0
17	all	all	0.1	0	0.1
18	all	one	0.1	0	0.1
19	one	all	0.1	0	0.1
20	one	one	0.1	0	0.1
21	all	all	0.1	0.1	0
22	all	one	0.1	0.1	0
23	one	all	0.1	0.1	0
24	one	one	0.1	0.1	0
25	all	all	0.1	0.1	0.1
26	all	one	0.1	0.1	0.1
27	one	all	0.1	0.1	0.1
28	one	one	0.1	0.1	0.1

Table 3.1 – A full DoE of binary parameters generated training simulations

3.2 Local model validation

Spot weld dynamical behaviors are elastoplastic. As we introduced in chapter 1, they are complex to predict because they are highly non-linear, especially near the rupture stage. We investigated whether the iDMD's local or global approach would be more appropriate to learn the ROM from the train set $\mathcal{T}$ and discussed the relevance of the selected approach.

Using manifold learning techniques, we found that the train set $\mathcal{T}$ lies o n a highly non-linear manifold. Thus, we chose the local rank-1 procedure (sec. 2.2) to construct the ROM. We built a dictionary of local models $\mathfrak{D}_b$ and defined the interpolation function for the prediction phase. We performed leave-one-out cross-validation and found the prediction error was below 1.5% (fig. 3.7). Those excellent results allowed us to validate the local linear approach.

In this section, we first introduce the selection of the local procedure, then we report the learning phase, and finally the validation of the ROM.

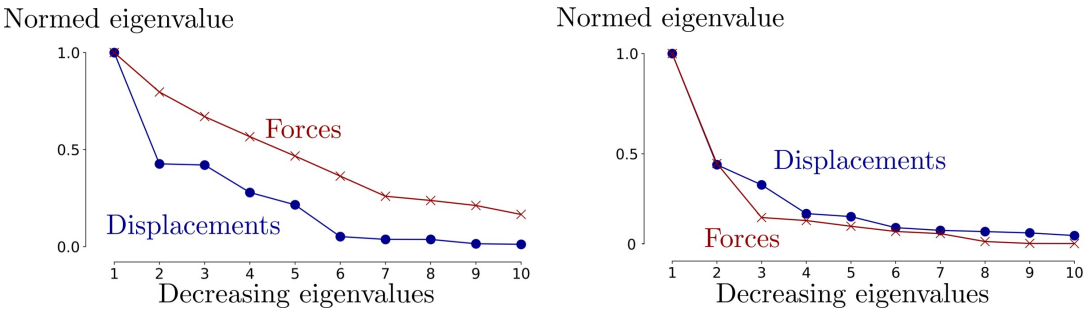
3.2.1 Motivating a local approach

MOR techniques, as introduced in the section 1.4, are based on the powerful but straightforward observation: in practice, a domain does not experience all the loading possibilities. Thus, the boundary displacements should belong to a manifold whose dimension is lower than the number of degrees of freedom N^{dof}. To approximate this manifold, we investigated whether the iDMD's local or global approach would be more appropriate. Indeed, when a global reduced-order model results in inaccurate predictions or increases computational time, depending on the size of the reduced basis, local approximations of the solution space are an appealing strategy.

To estimate the intrinsic dimension of the train set $\mathcal{T}$, we used two manifold learning techniques: the PCA and its non-linear counterpart the kernel Principal Component Analysis (kernel-PCA). We used the routines in scikit-learn [Ped+11].

We considered the displacements set $\mathcal{U}$ extracted from the train set $\mathcal{T}$,

$$\mathcal{U} = \{\mathbf{U}_b^1, ..., \mathbf{U}_b^{n_\mathcal{T}}\}$$



(a) PCA eigenvalues (b) kernel-PCA eigenvalues

Figure 3.4 – Extract of the 10th first eigenvalues using a linear and a non-linear manifold learning technique. The train set lied on a highly non-linear manifold of a reduced dimension.

and the forces set $\mathcal{F}$, also extracted from the train set $\mathcal{T}$:

$$\mathcal{F} = \{\mathbf{F}_b^1, ..., \mathbf{F}_b^{n_\mathcal{T}}\}$$

We applied the PCA on these two sets. The number of predominant modes allowed us to estimate the size of the reduced basis that could be extracted. We proceeded in the same way with kernel-PCA using the eigenvalues of the centered kernel matrix (computed with a sigmoïd kernel using the default parameters in [Ped+11]).

The kernel-PCA detected 5 predominant modes (out of $N_{dof} = 144$) on the displacements set $\mathcal{U}$ and the forces set $\mathcal{F}$ (fig. 3.4b), whereas the eigenvalues computed with the PCA smoothly decreased (fig. 3.4a). The PCA linear reduction performed poorly compared to the kernel-PCA's, thus we privileged the non-linear rank-1 approach for our use case. Additionally, those results illustrated the MOR main hypothesis, as the data lies on a manifold of a reduced dimension.

3.2.2 Rank-1 learning phase

Based on the learning phase of the rank-1 procedure presented in the section 2.2, we constructed a ROM predicting boundary forces $\mathbf{F}_b^i$ from boundary displacements $\mathbf{U}_b^i$. Thus, we built a dictionary $\mathfrak{D}_b$ of local models from the train set $\mathcal{T}$ and defined an interpolation function to predict the boundary forces $\hat{\mathbf{F}}_b$ for new boundary displacements

$\mathbf{U}_b \notin \mathcal{T}$:

$$\mathbf{F}_b = \mathcal{K}_{lr}\mathbf{U}_b$$

With $\mathcal{K}_{lr}$ the local model interpolated from the dictionary $\mathfrak{D}_b$.

We constructed the dictionary $\mathfrak{D}_b$ with $n_\mathcal{T}$ couples of column and row vectors respectively of size $(N_{dof} \times 1)$ and $(1 \times N_{dof})$ learnt from the train set $\mathcal{T}$ using the tensor decomposition (eq. 2.3):

$$\mathfrak{D}_b = \{(\mathbf{C}_b^1, \mathbf{R}_b^1), ..., (\mathbf{C}_b^{n_\mathcal{T}}, \mathbf{R}_b^{n_\mathcal{T}})\}$$

With the column vector $\mathbf{C}_b^i = \mathbf{F}_b^i$ from (eq. 2.5) and the row vector $\mathbf{R}_b^i = \frac{\mathbf{F}_b^i}{\mathbf{F_b^i}^T\mathbf{U}_b^i}$ from (eq. 2.6) whose Kronecker product would lead to $n_\mathcal{T}$ stiffness matrices. Each couple of column and row vector is labeled with the training couple $(\mathbf{U}_b^i, \mathbf{F}_b^i)$.

Once the dictionary was filled, we defined the interpolation function $\mathcal{I}$ so the rank-1 approach could interpolate models in $\mathfrak{D}_b$ (eq. 2.7) to predict boundary forces related to new boundary displacements $\mathbf{U}_b$:

$$\mathcal{K}_{lr}(\mathbf{U}_b) \approx \sum_{i=1}^{n_\mathcal{T}} \mathcal{K}_{lr}^i \mathcal{I}(\mathbf{U}_b),$$

We could have used various interpolation functions, we chose the nearest neighbor interpolation on the space of boundary displacements:

$$\mathcal{K}_{lr}^j(\mathbf{U}_b^j) = \mathcal{K}_{lr}^i(\text{argmin}_{\mathbf{U}_b^i} \|\mathbf{U}_b^j - \mathbf{U}_b^i\|_2),\ i = 0, ..., n_\mathcal{T}$$

Nearest neighbor methods find a number k of training vectors $\mathbf{U}_b^i$ closest in distance to the new input considered $\mathbf{U}_b$ and interpolate the related prediction from the k nearest neighbor outputs. Thus, one speaks of k-nearest neighbor learning. In general, the distance can be any metric measure. We chose the nearest neighbor interpolation ($k = 1$) with the standard Euclidean distance on this use case.

The naivest implementation of the neighborhood search is the brute force computation

of the distances between all pairs of points in the train set. For $n_{\mathcal{T}}$ samples in N_{dof} dimensions, this approach has a time complexity of $O(n_{\mathcal{T}} {N_{dof}}^2)$. Although brute force neighbor searches can be very efficient for small data sets, this approach quickly becomes infeasible as the number of samples $n_{\mathcal{T}}$ increases.

There are a variety of tree-based methods to address the computational inefficiencies of the brute force approach. These methods attempt to reduce the number of distance calculations by structuring the space. For example, if a new input $\mathbf{U}_b$ is far from the training sample $\mathbf{U}_b^1$ which is close to $\mathbf{U}_b^2$, we know $\mathbf{U}_b$ and $\mathbf{U}_b^2$ are far apart, without having to compute their distance explicitly. The time complexity of a nearest neighbor search then beneficially reduces to $O(n_{\mathcal{T}} N_{dof} \log(n_{\mathcal{T}}))$, or better. In this use case, we used the ball tree algorithm [Omo89] implemented in scikit-learn [Ped+11]. Ball trees partition data into series of nesting hyper-spheres, an efficient structure even in high dimensions (fig. 3.5).

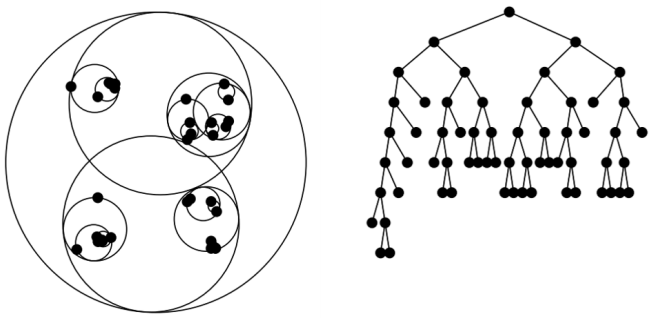


Figure 3.5 – A ball tree algorithm recursively divides the data into groups defined by a centroïd $C \in \mathbb{R}^{N_{dof}}$ and radius $r \in \mathbb{R}^+$, so that each point in the group lies within the hyper-sphere defined by r and C. The first leaf in the tree corresponds to the circle with the largest diameter. A single distance calculation between a new input and the centroïd determines a lower and an upper bound of the distance to a whole group. Figure from [Omo89].

3.2.3 Leave-one-out cross validation

After constructing the ROM in the learning phase, we assessed the generalization ability of the rank-1 approach, eventually to tune the interpolation function. Thus, we used cross-validation, the usual data resampling methods in machine learning [Ber19] for model validation.

Cross-validation divides the train set $\mathcal{T}$ into two subsets $\mathcal{T} = \mathcal{L} \cup \mathcal{V}$, where $\mathcal{L}$ is used to learn the dictionary of local models and $\mathcal{V}$ to validate the ROM. In general, the train and validation sets cross over in successive rounds, so the prediction of each couple $(\mathbf{U}_b^i, \mathbf{F}_b^i) \in \mathcal{T}$ is evaluated.

We used the Leave-One-Out Cross-Validation (LOOCV) [Ber19] where each couple $(\mathbf{U}_b^i, \mathbf{F}_b^i)$ served in turn as hold-out case for the validation set $\mathcal{V}$. Thus, the first validation set contained $(\mathbf{U}_b^1, \mathbf{F}_b^1)$, the second validation set contained $(\mathbf{U}_b^2, \mathbf{F}_b^2)$, and so on for the $n_\mathcal{T}$ couples in $\mathcal{T}$ (fig. 3.6).

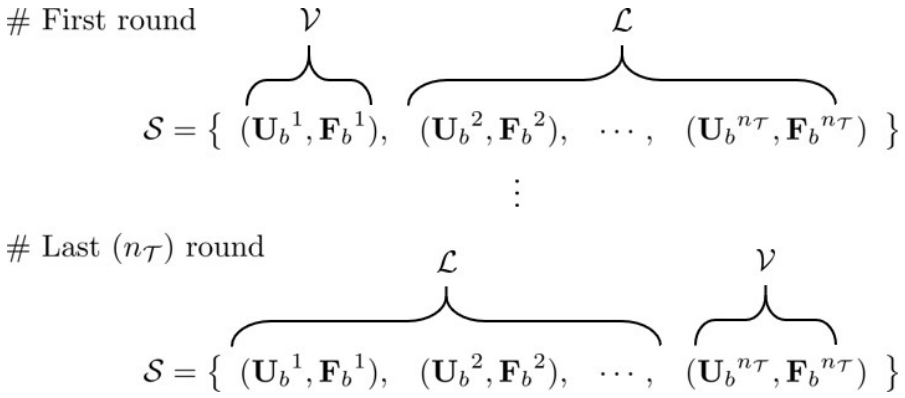


Figure 3.6 – Leave-One-Out Cross-Validation

At each round, we constructed a ROM from the set $\mathcal{L}$ using the rank-1 approach and predicted the boundary forces $\hat{\mathbf{F}}_b$ from the boundary displacements $\mathbf{U}_b$ in $\mathcal{V}$. We then computed the prediction error e which read:

$$e = \frac{\|\mathbf{F}_b - \hat{\mathbf{F}}_b\|_2}{\|\mathbf{F}_b\|_2} \tag{3.1}$$

With $\mathbf{F}_b$ the boundary forces of the couple in $\mathcal{V}$ sampled from the train set $\mathcal{T}$.

We found the relative error of prediction e was below 1.5% during the LOOCV (fig. 3.7). Thus the ability of the rank-1 approach to generalize is excellent when a rich dictionary is available. This low error also enabled us to validate the hypothesis of a local linear mapping between inputs and outputs made in the section 2.2. Moreover, those results illustrated the interpolation function $\mathcal{I}$ we defined is physically sound. Indeed, during the LOOCV, the local model selected associated to $\mathbf{U}_b$ was usually related to

the boundary displacement at the previous time step except for the displacements at the beginning of a simulation.

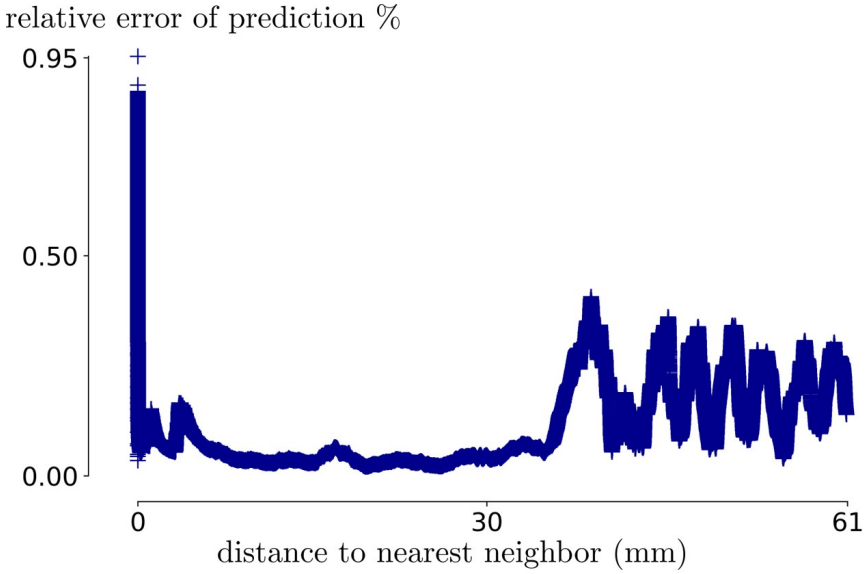


Figure 3.7 – Extract of the LOOCV on the training simulation (1). The rank-1 approach generalizes very well on a rich dictionary.

3.3 Prediction on the test set

Using the previously validated local procedure, we addressed the prediction of behaviors of spot welds that do not belong to the train set $\mathcal{T}$ to simulate an industrial problem. Thus, we defined a new numerical experiment and assessed the rank-1 performance as well as the local-global integration in VPS.

With a new DoE of five parameters (tab. 3.2), we computed 6 high-fidelity test simulations with VPS. We then used the local approach trained on the train set $\mathcal{T}$ to predict the test set $\mathcal{E}$. We found the ROM prediction error was lower than 10% before rupture and remained globally acceptable after. Besides, we validated the time integration of the local-global approach in VPS. Those encouraging results motivated the future integration of the local model embedded in the local-global approach in VPS to simulate spot weld behaviors during crashes.

This section introduces first the construction of the test set, then reports the prediction

of the test data, and finally addresses the time integration of the local-global approach in VPS.

3.3.1 Test set construction

To evaluate the performance of the rank-1 approach, we constructed a test set $\mathcal{E}$ with snapshots extracted from new simulations generated according to a test DoE (tab. 3.2). This test DoE contained the same loading and boundary conditions as the training DoE (tab. 3.1). However, the loading velocity was taken between the interval $[0, 0.15]$ (tab. 3.2).

Validation simulation	Closest training simulation	Recessed edges	Loaded edges	Loading magnitude (m/s)		
				V_x	V_y	V_z
1	1	all	all	0	0	0.15
2	25	all	all	0.07	0.07	0.07
3	25	all	all	0.07	0.05	0.09
4	26	all	one	0.07	0.05	0.09
5	27	one	all	0.07	0.05	0.09
6	28	one	one	0.07	0.05	0.09

Table 3.2 – Test DoE

We expected the prediction difficulty to vary depending on the test loading case as some were closest than others to the loading cases in the training DoE. For instance, the test loading case (1) was very similar to the training case (1). On the contrary, the test loading cases $(4, 5, 6)$ were less close to the training cases $(26, 27, 28)$ (tab. 3.2).

Using the test DoE, we ran 6 test simulations and extracted $n_{\mathcal{E}}$ couples:

$$\mathcal{E} = \{(\mathbf{U}_b^i, \mathbf{F}_b^i) \in \mathbb{R}^{N_{dof}} \times \mathbb{R}^{N_{dof}},\ i \in [\![0, n_{\mathcal{E}}]\!]\}$$

With:

- $n_{\mathcal{E}} = 300\,000$ snapshots
- the sampling period was the same as the training sampling period (3×10^{-4}ms).

3.3.2 Prediction with the rank-1 constructor

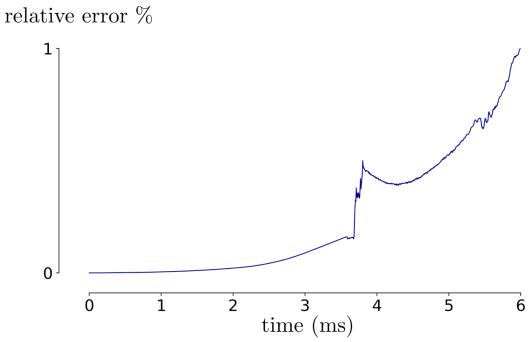
We analyzed the rank-1 prediction error on the test set $\mathcal{E}$. The ROM considered the dictionary $\mathfrak{D}_b$ of $n_{\mathcal{T}}$ stiffness matrices constructed from the train set $\mathcal{T}$. It predicted the boundary forces $\hat{\mathbf{F}}_b^j, j = 1, ..., n_{\mathcal{E}}$ related to the $n_{\mathcal{E}}$ boundary displacements $\mathbf{U}_b^j$ in $\mathcal{E}$ by selecting the adequate model in $\mathfrak{D}_b$.

$$\hat{\mathbf{F}}_b^j = \mathcal{K}_{lr}^j(\mathbf{U}_b^j)\mathbf{U}_b^j$$

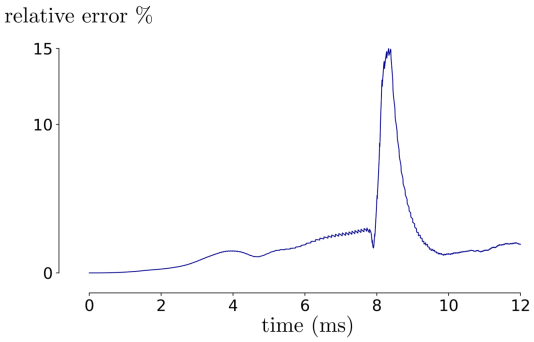
with $\mathbf{U}_b^j$ the test boundary displacements associated to $\mathbf{F}_b^j$.

To estimate the prediction accuracy of the ROM, we compared the reference boundary forces $\mathbf{F}_b^j$ in the test set $\mathcal{E}$ to their prediction $\hat{\mathbf{F}}_b^j$ (eq. 3.1). The relative errors (fig. 3.8) remained lower than 10% before the fracture starts, reaching maximum relative errors of 36% near rupture. This agreement was globally acceptable regarding the relative low number of local models in $\mathfrak{D}_b$ with respect to the complexity of the physical phenomena involved (figs. 3.9, 3.10, 3.11, 3.12, 3.13 and 3.14). Following those good results, ESI research team have started a more extensive study in a larger parametric space.

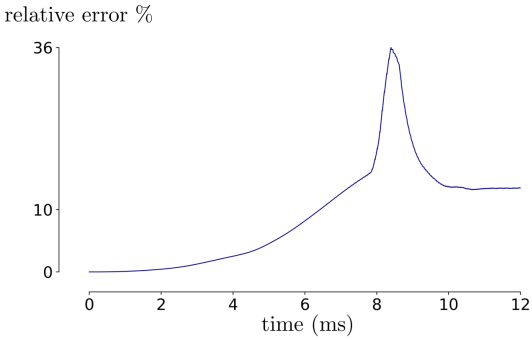
The small delays we observed in the predictions of the test simulations (n° 2, 3, 4, 5) appeared when similar boundary displacements $\mathbf{U}_b^i \sim \mathbf{U}_b^j$ in the train and test sets were associated to quite different training vector forces $\mathbf{F}_b^i$ and test vector forces $\mathbf{F}_b^j$, thus triggering the selection of a least adequate local model. We could define a variable that considers the mechanical history to enhance the accuracy of the predictions as introduced in chapter 2. However, in a crash test simulation, it is more important to predict how the spot welds rupture than when they rupture. Hence, these results are a convincing first step towards faster high-fidelity simulations.



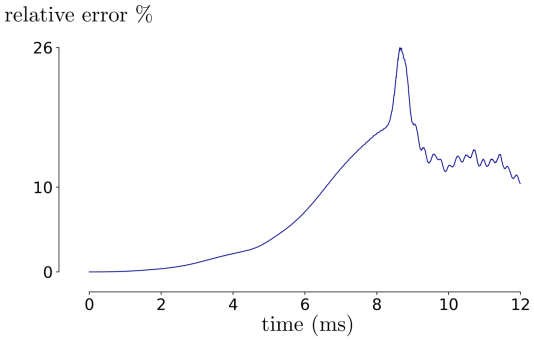
(a) Test simulation 1



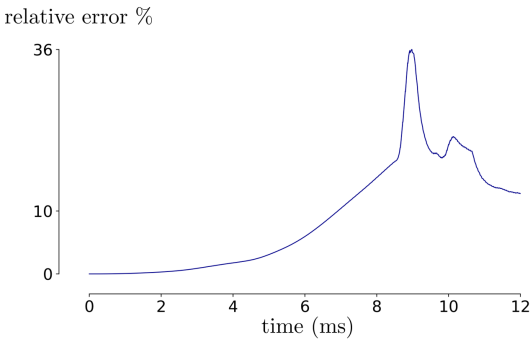
(b) Test simulation 2



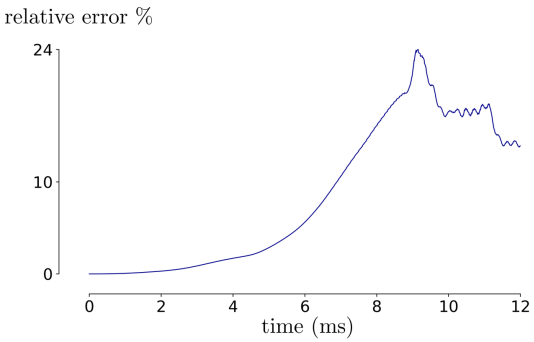
(c) Test simulation 3



(d) Test simulation 4



(e) Test simulation 5



(f) Test simulation 6

Figure 3.8 – The relative errors remained lower than 10% before the fracture starts. They slowly increase as cracks propagate in the spot weld before reaching maximum value of 35% when the complete rupture occurs.

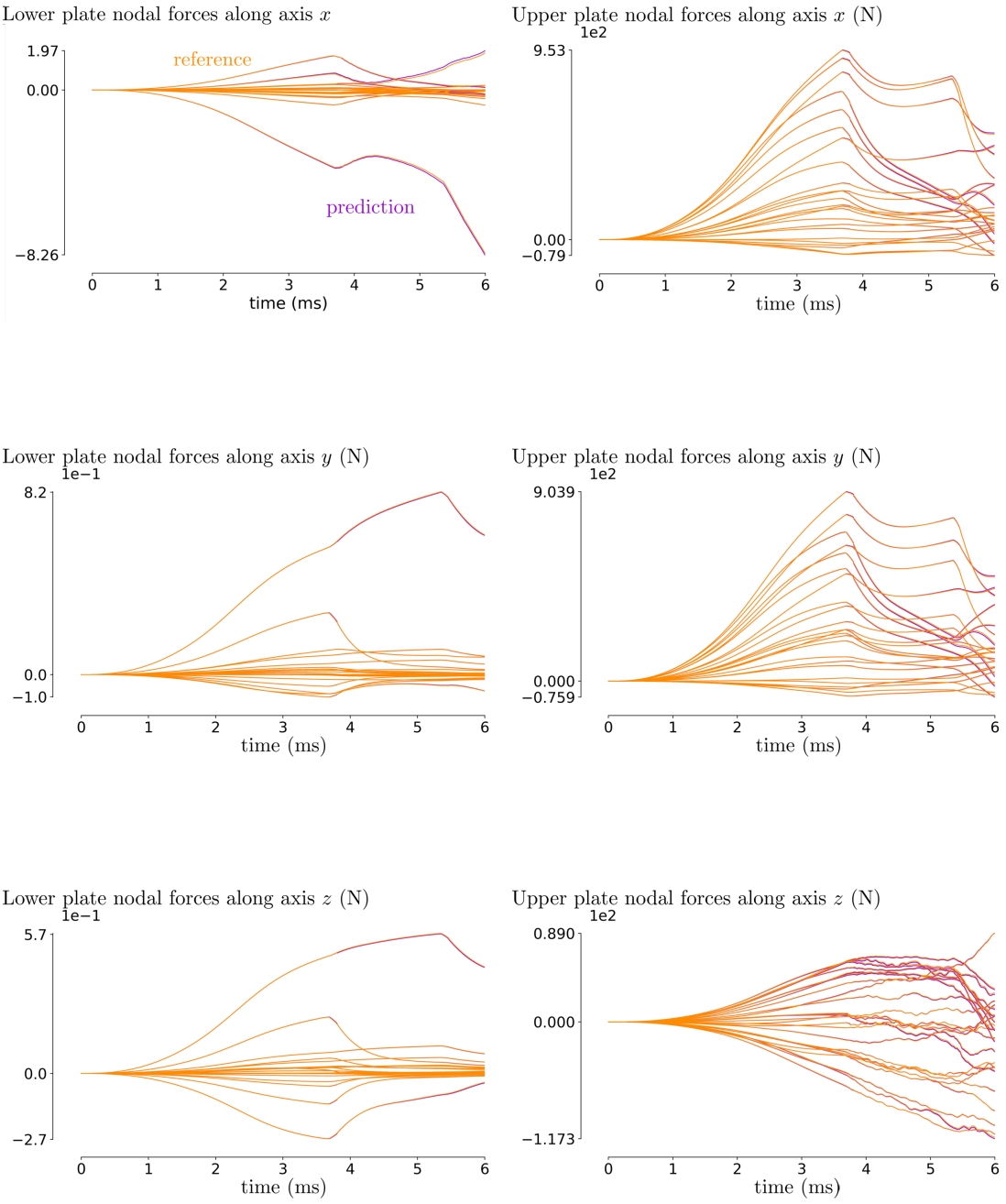


Figure 3.9 – Prediction of the boundary forces on the test simulation (1). Each violet and each orange curve represent the force according to one direction in a node. The spot weld started to tear at 3.8ms, and the complete rupture happened at 5.7ms. The prediction accuracy is excellent. The local models employed for the prediction were constructed from the training simulation (1), which is the closest parametrization in the training DoE.

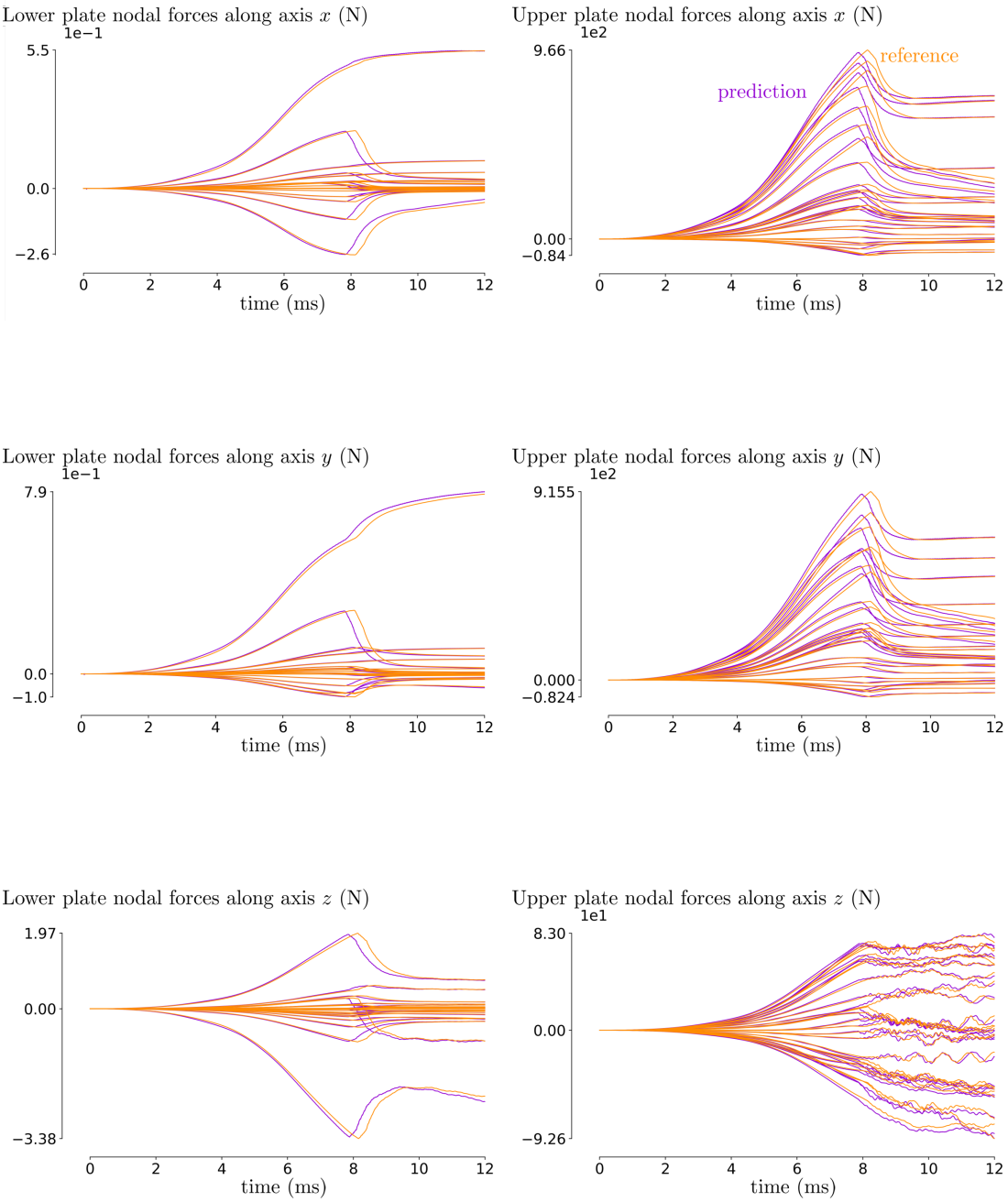


Figure 3.10 – Prediction of the boundary forces on the test simulation (2) with the rank-1 procedure. The predicted forces were globally in agreement with the reference forces. The local models employed for the prediction were constructed from the training simulation (25), which is the closest parametrization in the training DoE.

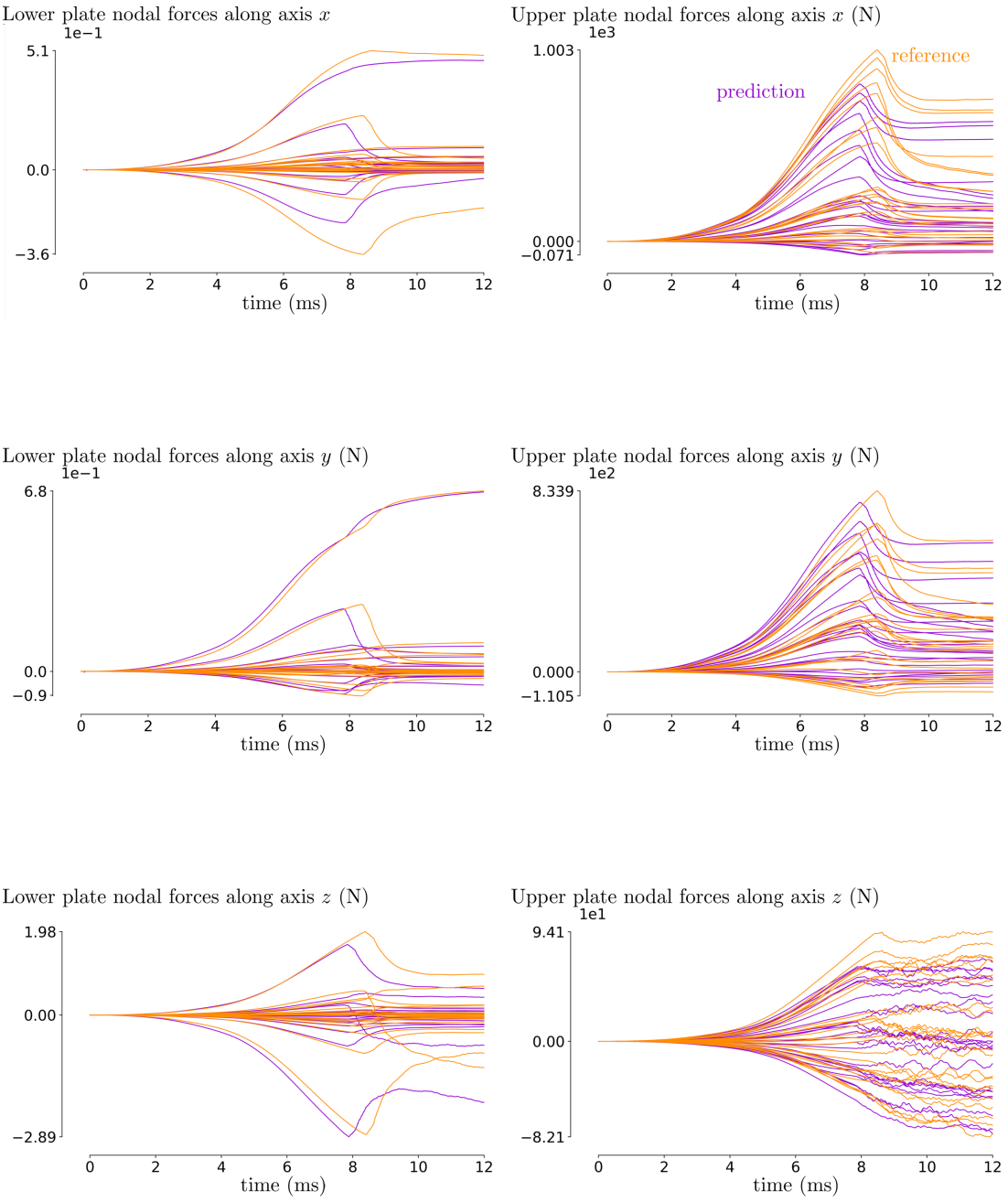


Figure 3.11 – The predicted forces were globally in agreement with the reference forces in the test simulation (3). The local models employed for the prediction were constructed from the training simulation (25), which is the closest parametrization in the training DoE. The parametrization of the previous test simulation (2) was closer to the training simulation (25) than this simulation and we observed more accurate predictions of the test simulation (2).

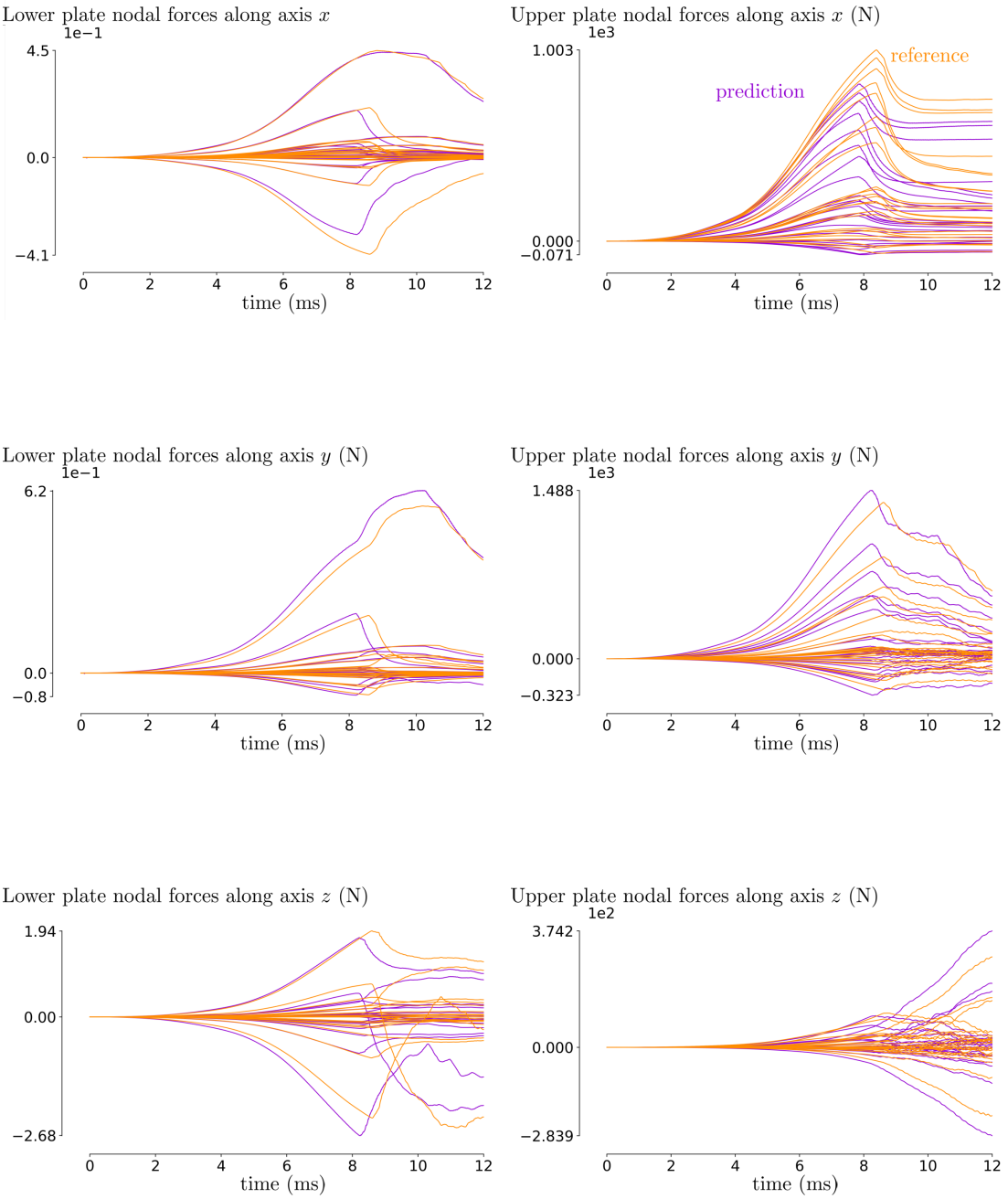


Figure 3.12 – The predicted forces were globally in agreement with the reference forces in the test simulation (4). We observed the rupture around 9ms. The local models employed for the prediction were constructed from the training simulation (26), which is the closest parametrization in the DoE.

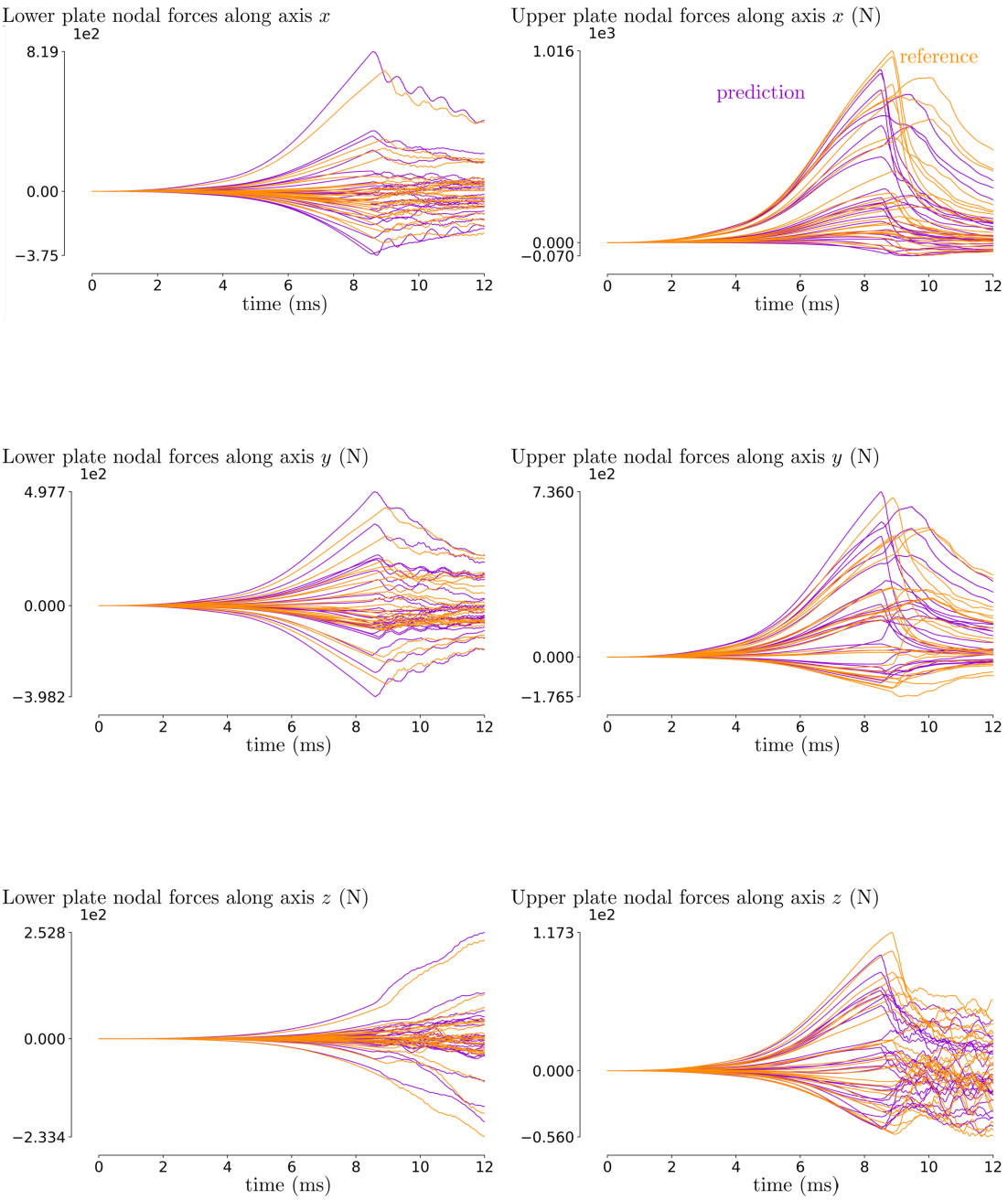


Figure 3.13 – The predicted forces are globally in agreement with the reference forces in the test simulation (5). We observe the rupture at 9ms. The local models employed for the prediction were constructed from the training simulation (27), which is the closest parametrization in the training DoE; thus, the interpolation function is physically sound.

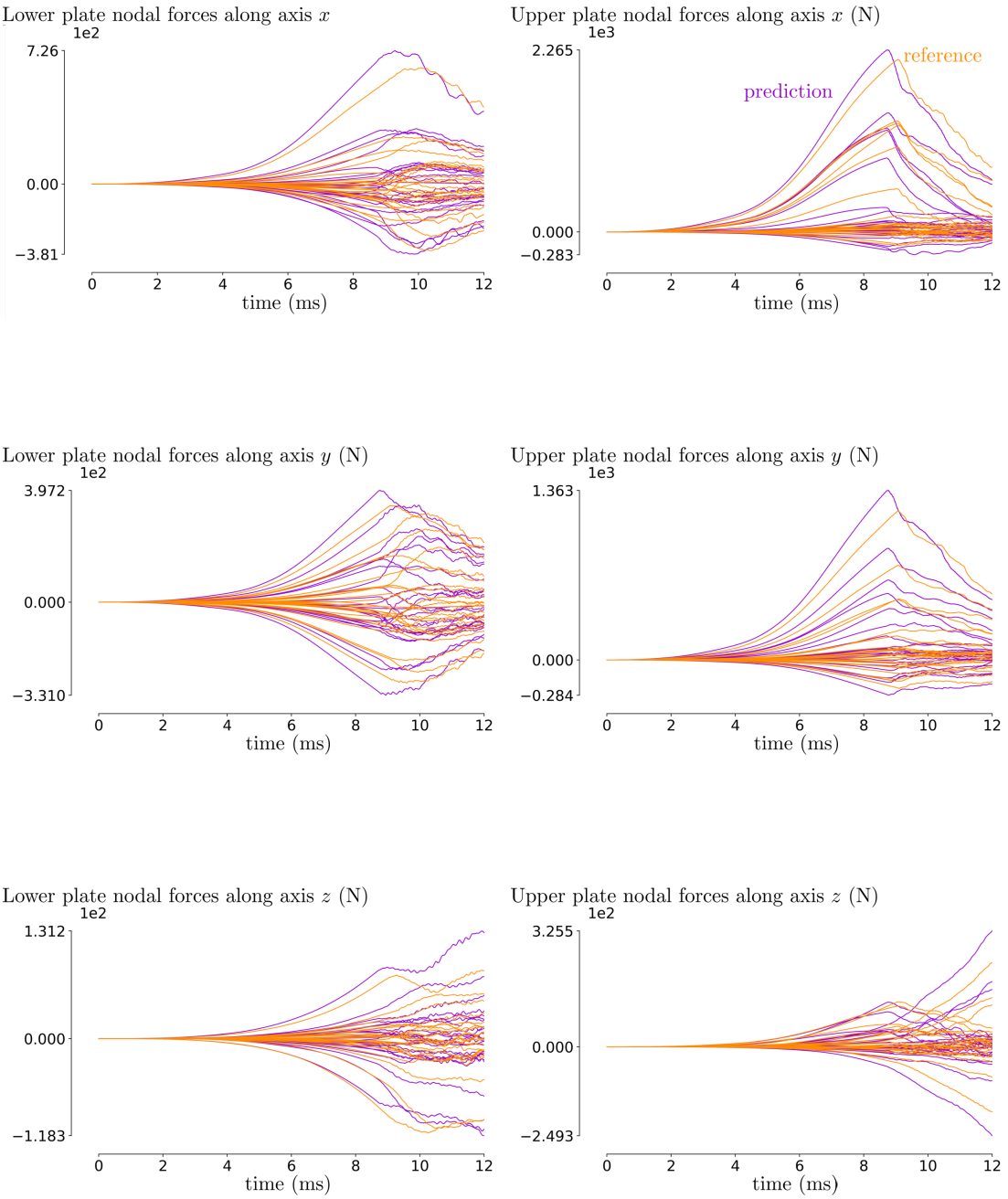


Figure 3.14 – The predicted forces were globally in agreement with the reference, the test simulation (6). We observed the rupture around 9ms. The local models employed for the prediction were constructed from the training simulation (28), which is the closest parametrization in the training DoE.

To evaluate the prediction time regarding the size of the dictionary $\mathfrak{D}_b$, we extracted local models from $\mathfrak{D}_b$ to construct a smaller dictionary $\mathfrak{D}_s$. Then, we used $\mathfrak{D}_s$ to predict the boundary forces in the test set $\mathcal{E}$ and measured CPU time. We extracted the models regularly and tried different sizes for $\mathfrak{D}_s$ with the rank-1 approach using 4 processors. Hopefully, the results (tab. 3.3) showed that the size of the dictionary did not influence the prediction time, perhaps because of the manifold's intrinsic structure.

Number of local models in $\mathfrak{D}_s$	Average prediction time on $\mathcal{V}$ (min)
1 400 000	7.5
700 000	6.7
280 000	7.3
140 000	6.9
70 000	6.1
56 000	6.1

Table 3.3 – The size of the dictionary does not strongly influence the prediction time

We proposed different strategies to reduce the prediction time if in an other use case the dictionary size had an influence:

- reduce the size of the dictionary. The snapshots sampled after the complete rupture are not relevant and could be withdrawn from the dictionary. Moreover, the snapshot's sampling can be parametrized: a finer sampling could be performed near the rupture stage, while we could perform a coarser one before.
- restrict the interpolation function domain. The predictions were performed using local models coming each time from the simulation with the closest parametrization.

3.3.3 Local-global time scheme integration

As the rank-1 constructor showed promising results, we addressed the time integration using the local-global approach in VPS. We considered the global domain $\tilde{\Omega}$ and the first loading case in the test DoE. We applied the loading on the domain boundary $\partial\Omega$, and on the local-to-global boundary $\partial\omega$ we applied boundary forces $\mathbf{F}_b^i$ predicted with the local approach (fig. 3.15).

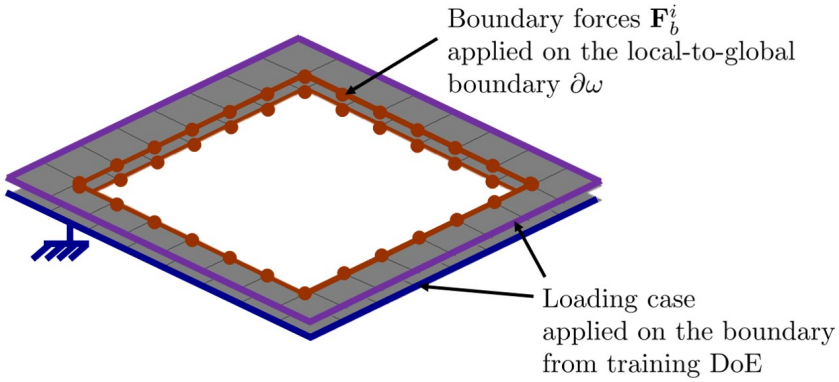
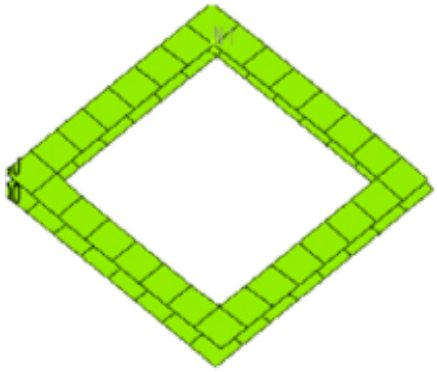


Figure 3.15 – The effects of the local domain are applied as boundary conditions on $\partial\omega$

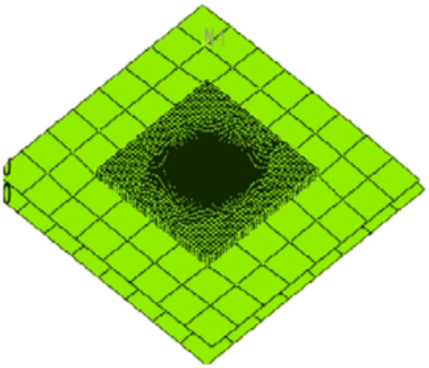
Algorithm 3: Global-local validation

if *i=1* **then**
  Compute the nodal accelerations in the domain $\tilde{\Omega}$ from the initial conditions (null displacements and velocities).
  Integrate acceleration to compute velocities and displacements at $t^{i=1}$.
end
for $1 \leq i \leq n_f$ **do**
  Extract the boundary displacements $\mathbf{U}_b^i$.
  Predict the boundary forces $\mathbf{F}_b^i$ related to $\mathbf{U}_b^i$ with the rank-1 approach.
  Compute the nodal accelerations in the domain $\tilde{\Omega}$ by solving the momentum balance equation: $\tilde{\mathcal{M}}\ddot{\mathbf{U}}^i = \tilde{\mathbf{F}}_{int}^i + \tilde{\mathbf{F}}_{ext}^i + \mathbf{F}_b^i$ where $\tilde{\bullet}$ refers to quantities defined in $\tilde{\Omega}$ and $\mathbf{F}_b^i$ traduces the local domain effects on $\partial\omega$;
  Integrate acceleration to compute velocities and displacements at time $t + \Delta t$;
end

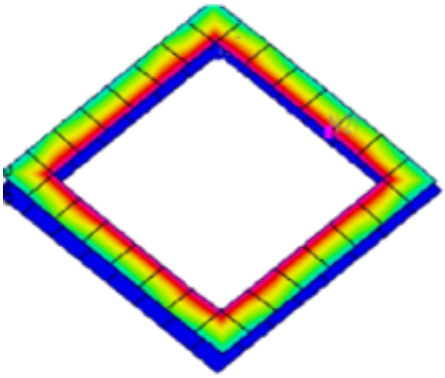
We used the algorithm 3 for the validation of local-global integration where at each time step, boundary forces are applied on the boundary $\partial\omega$. This integration considered the impacts of the domain decomposition in VPS and the effects of the ROM. We only used the forces and the displacements and excluded the momentum and the rotations, as in the train set construction (sec. 3.1.2). Despite this restriction, the results were in good agreement with the high-fidelity simulation (fig. 3.16). Thus, ESI research team have started a larger implementation with moments and rotations.



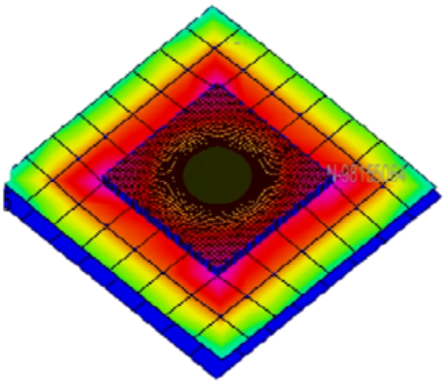
(a) Global domain $\tilde{\Omega}$ at $t^i = 0$ms



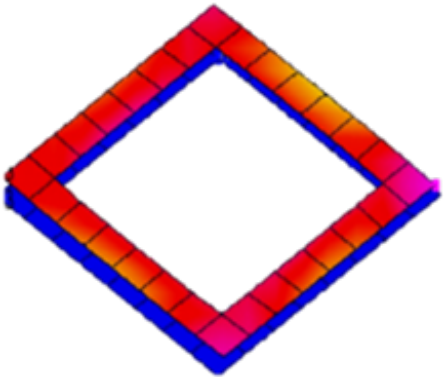
(b) Domain Ω at $t^i = 0$ms



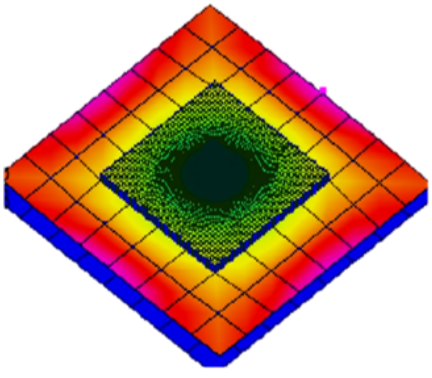
(c) Global domain $\tilde{\Omega}$ at $t^i = 2$ms



(d) Domain Ω at $t^i = 2$ms



(e) Global domain $\tilde{\Omega}$ at $t^i = 5$ms



(f) Domain Ω at $t^i = 5$ms

Figure 3.16 – The displacements field is in the same order of magnitude between the local-global approach on the left and the meso-scale model on the right. Moreover, the time step in the explicit scheme using the local-global approach is of the order of magnitude of the time step in the global domain resulting in a reduced computational time in the global domain $\tilde{\Omega}$.

Chapter 4

Conclusion

4.1 Reminder of the issue

We presented in chapter 1 two needs of automotive manufacturers. First, to perform only confirmation crash tests to reduce the design cost of vehicle projects. Second, to conduct intensive virtual testing to determine the optimal weight of car parts. Indeed, engineers' strategy to limit consumption on both thermal and electric vehicles is to design lighter vehicles.

There is a technical answer to these two needs: high-fidelity crash simulation that run in less than a day. Indeed, crash simulation has become a fundamental design tool beyond its initial use in the safety domain. After showing that moving towards high-fidelity crash test simulations was necessary for virtual testing, we highlighted why fast simulations were required.

To compute crash test simulations closer to reality, engineers integrate local models that account for the dynamic behaviors of small assembly elements. Indeed, these small components are critical regions often at the parentage of structural weakening during a car crash. This thesis focuses on spot welds in crash test simulation as they are widely used in the automotive industry: a car contains on average 4000 of them.

Although the complexity of the physical phenomena involved in making a spot weld is widely recognized, resulting in many models of spot welds, accurate models allowing for rapid crash test simulations are the subject of few work. In our state of the art, we have highlighted that the main macro and mesoscale models of spot welds predict with good agreement the experimental results. Therefore, the main difficulty remains the computation time, despite the generalization of HPC.

We have shown that some approaches are an improvement for fast high-fidelity crash test simulations in an industrial context. However, no integration of spot weld models into a crash test solver so far meets the accuracy and speed requirements of manufacturers as high-fidelity crash test simulations become extremely time consuming when taking into account local component behaviors. Indeed, the usual explicit integration used for fast dynamics suffers from the tiny time steps and the extremely fine meshes required to adequately represent 4000 spot welds. We therefore investigated MOR techniques to construct a ROM of a mesoscale spot weld model. Those techniques

are interesting to improve the accuracy of a crash test model at a low computational cost as they exploit the generally low dimensionality of the loading space of a physical phenomenon and the correlations in the loading .

4.2 Work conducted

We inspired ourselves from the DMD and proposed a new ROM technic in chapter 2, the iDMD. This procedure contains two constructors, a global linear and a local non-linear one. The local rank-1 approach is suited for the reduction of the highly non-linear spot weld behaviors. This ROM constructs a dictionary of local linear rank-1 models from $n_{\mathcal{T}}$ couples of displacements and forces $(\mathbf{U}^i, \mathbf{F}^i)$ sampled from high-fidelity simulations.

$$\mathfrak{D} = \left\{ \mathcal{K}_{lr}^1 , ..., \mathcal{K}_{lr}^{n_{\mathcal{T}}} \right\} \tag{4.1}$$

In the prediction phase, the function $\mathcal{I}$ interpolates models in $\mathfrak{D}$ to predict the forces $\hat{\mathbf{F}}$ related to a new sampled displacements $\mathbf{U}$:

$$\hat{\mathbf{F}} = \sum_{i=1}^{n_{\mathcal{T}}} \mathcal{K}_{lr}^i \mathcal{I}(\mathbf{U}) \ = \mathcal{K}_{lr}\mathbf{U} \tag{4.2}$$

To couple the spot weld ROM into the crash solver, we revisited mechanical condensation and called our procedure the local-global approach. We proposed to construct a condensed ROM with the iDMD. The ROM learns the behaviors of the interface, called local-to-global boundary, separating the local domain (the spot weld), where the condensation applies, from the global one (the car) to perform the coupling of the local model to the global one. Thus, the ROM can infer forces at the nodes on the boundary between the two domains for any displacement applied on the global domain.

We illustrated the rank-1 reduction technique and the local-global approach on a use case on spot welds in chapter 3. We considered a spot weld ROM joining two sheets and discussed the impacts of the local-global integration in VPS, the ESI crash test solver. We defined loading close to unitary loading encountered in mechanical tests such as tensile and peeling tests. For 6 high-fidelity test simulations, the rank-1 approach reduced the computational time on 16 processors from 1h30 to less than 8 min. The

relative error of prediction was lower than 10% before rupture and remained globally acceptable during crack propagation. We then evaluated the local-global approach: we tested a coupling with the iDMD and this integration framework demonstrated promising results for the crash solver as the condensed ROM operated at the global structure time step.

4.3 Conclusion on the issue

We investigated the integration of ROMs of spot welds instead of mesoscale models in crash solvers to speed up calculation time. Undoubtedly, the ROM of the boundary constructed with the rank-1 procedure performed more rapidly than the reference mesoscale model. In our experiment, the boundary forces were computed at the global structure time step. The accuracy of the reduced model is excellent as long as the damage is not advanced and remains acceptable during crack propagation. The relative error was about 10% before rupture and the rupture time was also accurately predicted. When a rich dictionary was available, we obtained excellent results including during the crack propagation. However, when addressing complex crash loading, the size of the dictionary might increase tremendously. It could generate memory issues and slow down the prediction phase. Indeed, the prediction time of the interpolation function $\mathcal{I}$ we used, the nearest neighbor interpolation, depends on the size of the dictionary. Moreover, populating the vast loading space to construct a rich dictionary could require extremely intensive computations. However, engineers design cars to avoid specific rupture modes, hence the loading space can be at the end of a reasonable size.

In this thesis, we proposed an integration method for reduced models into a crash solver. When ROM are applied to model rich and localized behaviors of spot welds within a global domain, the simulation of the global domain can be performed by keeping the local domain effects but without explicitly resolving the local problem. The main advantages of such a procedure are, first, the reduction of the number of degrees of freedom as the global model does not consider anymore the fine 3D discretizations that usual accurate spot weld models imply. Second, the increase of the time step in explicit integration. Indeed, the integration stability is no longer governed

by the smallest element's size used for describing the solution at the spot weld level. Of course, this discussion can be extended to any local fine discretization in a coarser mesh.

4.4 Limits

It is still too early to conclude on the proposed methodology's accuracy and execution time performance when applied to industrial crash test problems. The present work illustrated the local-global performances in a simple use case and succeeded in establishing the condensed ROM of a spot weld. However, to conclude on its industrial applicability, the model should be enriched: (i) by enlarging the sampling used in the model training; (ii) by considering a parametrization of the displacements to tackle non-monotonic loading; and (iii) by setting-up the local-global methodology with a fine study of the error propagation in the global model and the CPU time savings. We expect the performances to be significant because of the high number of spot welds (thousands in the car structure) and the ultra-fast evaluation of the constructed ROM if the size of the local model dictionary remains acceptable.

4.5 Perspectives and ongoing work

We defined a work plan comprising several tasks to evaluate the industrialization of our approach. The MOR consortium will conduct those tasks. This first task is to perform many crash tests simulations with macroscale models of spot welds to define the loading representative of real crash tests. This task will provide both an estimate of the actual problem dimension, the dictionary size, and the typical loading spot weld encounter in crash tests. The second task concerns the validation of the approach by comparing the results of crash test calculations carried out with the mesoscale spot weld models with the same calculations carried out with the reduced spot weld models. We must verify that the ROM do not induce a divergence of the crash calculations. If the results are conclusive, an industrial integration in VPS will be conducted, leading towards high-fidelity simulations.

Bibliography

[AEB06] M. Aharon, M. Elad, and A. Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation". In: *IEEE Transactions on Signal Processing* 54.11 (2006), pp. 4311–4322.

[ANR74] N. Ahmed, T. Natarajan, and K.R. Rao. "Discrete Cosine Transform". In: *IEEE Transactions on Computers* C-23.1 (1974), pp. 90–93.

[AK18] N. Altman and M. Krzywinski. "The curse(s) of dimensionality". In: *Nature Methods* 15.6 (2018), pp. 399–400.

[Amm+06] A. Ammar et al. "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids". In: *Journal of Non-Newtonian Fluid Mechanics* 139.3 (2006), pp. 153–176.

[Amm+07] A. Ammar et al. "A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids: Part II: Transient simulation using space-time separated representations". In: *Journal of Non-Newtonian Fluid Mechanics* 144.2 (2007), pp. 98–121.

[AF08] D. Amsallem and C. Farhat. "Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity". In: *AIAA Journal* 46.7 (2008), pp. 1803–1813.

[AZF12] D. Amsallem, M. J. Zahr, and C. Farhat. "Nonlinear model order reduction based on local reduced-order bases". In: *International Journal for Numerical Methods in Engineering* 92.10 (2012), pp. 891–916.

[ASG01] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. “A survey of model reduction methods for large-scale systems”. In: *Contemporary Mathematics* 280 (2001), pp. 193–219.

[Aut12] American Welding Society. Committee on Automotive Welding. *Test Methods for Evaluating the Resistance Spot Welding Behavior of Automotive Sheet Steel Materials*. American Welding Society, 2012.

[Bad+17] A. Badías et al. “Local proper generalized decomposition: Local proper generalized decomposition”. In: *International Journal for Numerical Methods in Engineering* 112.12 (2017), pp. 1715–1732.

[BSK91] K. S. Ball, L. Sirovich, and L. R. Keefe. “Dynamical eigenfunction decomposition of turbulent channel flow”. In: *International Journal for Numerical Methods in Fluids* 12.6 (1991), pp. 585–604.

[Bar+04] M. Barrault et al. “An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations”. In: *Comptes Rendus Mathematique* 339.9 (2004), pp. 667–672.

[BTG94] D. Benoist, Y. Tourbier, and S. Germain-Tourbier. *Plans d’expériences: construction et analyse*. Paris: Lavoisier, 1994. 700 pp.

[BHS94] G. Béranger, G. Henry, and G. Sanz. *Le livre de l’acier*. Paris: Technique & documentation Lavoisier, 1994.

[Ber19] D. Berrar. “Cross-Validation”. In: *Encyclopedia of Bioinformatics and Computational Biology*. Elsevier, 2019, pp. 542–545.

[BLC14] B. Bognet, A. Leygue, and F Chinesta. “Separated representations of 3D elastic solutions in shell geometries”. In: *Advanced Modeling and Simulation in Engineering Sciences* 1.1 (2014), p. 4.

[Bor08] R. de Borst. “Challenges in computational materials science: Multiple scales, multi-physics and evolving discontinuities”. In: *Computational Materials Science* 43.1 (2008), pp. 1–15.

[BAC19] D. Borzacchiello, J. V. Aguado, and F. Chinesta. “Non-intrusive Sparse Subspace Learning for Parametrized Problems”. In: *Archives of Computational Methods in Engineering* 26.2 (2019), pp. 303–326.

[Caz93] R. Cazes. “Soudage par résistance”. In: *Techniques de l’ingénieur* (1993), p. 22.

[Caz97] R. Cazes. “Soudage automatique”. In: *Techniques de l’ingénieur* (1997), p. 18.

[Cha02] Y. J. Chad. “Failure of Spot Weld: A Competition between Crack Mechanics and Plastic Collapse”. In: *Recent Advances in Experimental Mechanics: In Honor of Isaac M. Daniel*. Dordrecht, 2002, pp. 245–256.

[Cha03] Y.J. Chao. “Failure mode of spot welds: interfacial versus pullout”. In: *Science and technology of welding and joining* 8.2 (2003), pp. 133–137.

[Cha00] A. Chatterjee. “An introduction to the proper orthogonal decomposition”. In: *Current Science* 78.7 (2000), pp. 808–817.

[CS10] S. Chaturantabut and D. C. Sorensen. “Nonlinear Model Reduction via Discrete Empirical Interpolation”. In: *SIAM Journal on Scientific Computing* 32.5 (2010), pp. 2737–2764.

[CD00] W. Chen and X. Deng. “Performance of shell elements in modeling spot-welded joints”. In: *Finite Elements in Analysis and Design* 35.1 (2000), pp. 41–57.

[CAC10a] F. Chinesta, A. Ammar, and E. Cueto. “On the use of proper generalized decompositions for solving the multidimensional chemical master equation”. In: *European Journal of Computational Mechanics* 19.1 (2010), pp. 53–64.

[CAC10b] F. Chinesta, A. Ammar, and E. Cueto. “Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models”. In: *Archives of Computational Methods in Engineering* 17.4 (2010), pp. 327–350.

[CKL13] F. Chinesta, R. Keunings, and A. Leygue. *The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer*. Springer Science & Business Media, 2013. 127 pp.

[Chi+13] F. Chinesta et al. “PGD-Based Computational Vademecum for Efficient Design, Optimization and Control”. In: *Archives of Computational Methods in Engineering* 20.1 (2013), pp. 31–59.

[Chi+17] F. Chinesta et al. “Model Reduction Methods”. In: *Encyclopedia of Computational Mechanics Second Edition*. Ed. by E. Stein, R. de Borst, and T. J R Hughes. John Wiley & Sons, Ltd, 2017, pp. 1–36.

[CGP13] A. Coussement, O. Gicquel, and A. Parente. “MG-local-PCA method for reduced order combustion modeling”. In: *Proceedings of the Combustion Institute* 34.1 (2013), pp. 1117–1123.

[CGA16] E. Cueto, D. González, and I. Alfaro. *Proper Generalized Decompositions: An Introduction to Computer Implementation with Matlab*. Springer, 2016. 103 pp.

[Dan09] S. Dancette. “Comportement mécanique des soudures par points : mécanismes et stratégies de prédiction dans le cas des tôles en acier pour automobile”. PhD thesis. 2009.

[Dan+20] T. Daniel et al. “Model order reduction assisted by deep neural networks (ROM-net)”. In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), p. 16.

[DCE17] S. E. Davis, S. Cremaschi, and M. R. Eden. “Efficient Surrogate Model Development: Optimum Model Form Based on Input Function Characteristics”. In: *Computer Aided Chemical Engineering*. Ed. by Antonio Espuña, Moisès Graells, and Luis Puigjaner. Vol. 40. 27 European Symposium on Computer Aided Process Engineering. Elsevier, 2017, pp. 457–462.

[DG00] H. Di Fant-Jaeckels and A. Galtier. “Fatigue lifetime prediction model for spot welded structures”. In: *Revue de Métallurgie* 97.1 (2000), pp. 83–96.

[DZH17] P. Díez, S. Zlotnik, and A. Huerta. “Generalized parametric solutions in Stokes flow”. In: *Computer Methods in Applied Mechanics and Engineering* 326 (2017), pp. 223–240.

[Din+06] S. Dinçer et al. *A Comparative Study on the Finite Element Models for Spot Welds and Their Verification*. SAE Technical Paper 2006-01-0590. Warrendale, PA: SAE International, Apr. 3, 2006.

[Don+05] S. Donders et al. “The Effect of Spot Weld Failure on Dynamic Vehicle Performance”. In: *Sound and vibration* (2005), p. 10.

[Dum+17] A Dumon et al. “Adaptive local weld models in advanced high-strength steels for use in full-vehicule crash simulation”. In: Nafems world congress. 2017, p. 12.

[Dun17] Dunya. https://www.dunya.com/sektorler/otomotiv/avrupa-otomotiv-pazari-yuzde-45-buyudu-haberi-374676. 2017.

[Dur+15] S. Duraffourg et al. “New spot weld fatigue criteria and fatigue data editing technique”. In: *Procedia Engineering* 133 (2015), pp. 433–453.

[Enn14] P.J. Ennis. “Ferritic and martensitic steels for power plants”. In: *Structural Alloys for Power Plants*. Elsevier, 2014, pp. 188–220.

[Eur20a] Euro NCAP. *Euro NCAP Assessment Protocol - Overall Rating - v9.0.1*. Tech. rep. 2020.

[Eur20b] Euro NCAP. *Test d’impact latéral contre un poteau*. https://www.euroncap.com/fr/securite-des-vehicules/la-notation-en-detail/protection-des-adultes/collision-laterale/test-poteau/. 2020.

[Eur21a] Euro NCAP. *Dernières notations de sécurité*. Le site officiel du Programme Européen pour l’Evaluation d’Automobiles Neuves. https://www.euroncap.com/fr/notations-et-recompenses/dernieres-notations-de-securite. 2021.

[Eur21b] Euro NCAP. *How To Read The Stars*. The Official Site of The European New Car Assessment Programme. https://www.euroncap.com/en/about-euro-ncap/how-to-read-the-stars/. 2021.

[Fan+00] J Fang et al. “Weld modeling with MSC/Nastran”. In: *Second MSC Worldwide Automotive User Conference, Dearborn, MI*. Citeseer. 2000.

[Fey99] F. Feyel. “Multiscale FE2 elastoviscoplastic analysis of composite structures”. In: *Computational Materials Science* 16.1 (1999), pp. 344–354.

[GKB10] M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. “Multi-scale computational homogenization: Trends and challenges”. In: *Journal of Computational and Applied Mathematics* 234.7 (2010), pp. 2175–2182.

[GH07] E. Grosu and I. Harari. “Stability of semidiscrete formulations for elastodynamics at small time steps”. In: *Finite Elements in Analysis and Design* 43.6 (Apr. 2007), pp. 533–542.

[Gst20] E. Gstalter. “Réduction d’ordre de modèle de crash automobile pour l’optimisation masse / prestations”. PhD thesis. 2020.

[Gst+20] E. Gstalter et al. “Towards new methods for optimization study in automotive industry including recent reduction techniques”. In: *Advanced Modeling and Simulation in Engineering Sciences* 7.1 (2020), p. 17.

[HH92] E. R. Henry and J. Hofrichter. “Singular value decomposition: Application to analysis of experimental data”. In: *Methods in Enzymology*. Vol. 210. Numerical Computer Methods. Academic Press, 1992, pp. 129–192.

[Hua+20] W. Huang et al. “Nonlinear Optimization of Orthotropic Steel Deck System Based on Response Surface Methodology”. In: *Research* 2020 (2020), pp. 1–22.

[Ibá+18] R. Ibáñez et al. “A Multidimensional Data-Driven Sparse Identification Technique: The Sparse Proper Generalized Decomposition”. In: *Complexity* 2018 (2018), pp. 1–11.

[Ibá+19] R. Ibáñez et al. “A local multiple proper generalized decomposition based on the partition of unity”. In: *International Journal for Numerical Methods in Engineering* 120.2 (2019), pp. 139–152.

[Kah+14] P. Kah et al. “Welding of Ultra High Strength Steels”. In: *Advanced Materials Research* 849 (2014), pp. 357–365.

[KL97] N. Kambhatla and T. K. Leen. “Dimension Reduction by Local Principal Component Analysis”. In: *Neural Computation* 9.7 (1997), pp. 1493–1516.

[KPB12] P. Kerfriden, J. C. Passieux, and S. P. A. Bordas. "Local/global model order reduction strategy for the simulation of quasi-brittle fracture". In: *International Journal for Numerical Methods in Engineering* 89.2 (2012), pp. 154–179.

[Ker+13] P. Kerfriden et al. "A partitioned model order reduction approach to rationalise computational expenses in nonlinear fracture mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 256 (2013), pp. 169–188.

[KT14] N. Khandoker and M. Takla. "Tensile strength and failure simulation of simplified spot weld models". In: *Materials & Design (1980-2015)* 54 (2014), pp. 323–330.

[KO16] T. Kirchdoerfer and M. Ortiz. "Data-driven computational mechanics". In: *Computer Methods in Applied Mechanics and Engineering* 304 (2016), pp. 81–101.

[Kor+12] J. Korta et al. "Multi-Objective Optimization of a Car Body Structure". In: *SAE International Journal of Passenger Cars - Mechanical Systems* 5.3 (2012), pp. 1143–1152.

[KKW08] R Kuziak, Rudolf Kawalla, and Sebastian Waengler. "Advanced high strength steels for automotive industry". In: *Archives of civil and mechanical engineering* 8.2 (2008), pp. 103–117.

[LPN10] P. Ladevèze, J.-C. Passieux, and D. Néron. "The LATIN multiscale computational method and the Proper Generalized Decomposition". In: *Computer Methods in Applied Mechanics and Engineering* 199.21 (2010), pp. 1287–1296.

[Lam+10] H. Lamari et al. "Routes for Efficient Computational Homogenization of Nonlinear Materials Using the Proper Generalized Decompositions". In: *Archives of Computational Methods in Engineering* 17.4 (2010), pp. 373–391.

[Lam07] E. H. J. Lamouroux. "Modélisation détaillée des soudures par point et laser pour simuler la rupture des assemblages automobiles". PhD thesis. 2007.

[Le +18] Y. Le Guennec et al. “A parametric and non-intrusive reduced order model of car crash simulation”. In: *Computer Methods in Applied Mechanics and Engineering* 338 (2018).

[LD18] H. Li and C. A. Duarte. “A two-scale generalized finite element method for parallel simulations of spot welds in large structures”. In: *Computer Methods in Applied Mechanics and Engineering* 337 (2018), pp. 28–65.

[Lim+19] V. Limousin et al. “Advanced model order reduction and artificial intelligence techniques empowering advanced structural mechanics simulations: application to crash test analyses”. In: *Mechanics & Industry* 20.8 (2019), p. 804.

[LLP06] P. -C. Lin, S. -H. Lin, and J. Pan. “Modeling of failure near spot welds in lap-shear specimens based on a plane stress rigid inclusion analysis”. In: *Engineering Fracture Mechanics* 73.15 (2006), pp. 2229–2249.

[Lin+02] S.-H. Lin et al. “Failure loads of spot welds under combined opening and shear static loading conditions”. In: *International Journal of Solids and Structures* 39.1 (2002), pp. 19–39.

[Loè60] M. Loève. *Probability theory*. 3rd edition. Van Nostrand, 1960. 685 pp.

[LR08] V. H. López-Cortéz and F. A. Reyes-Valdés. “Understanding Resistance Spot Welding of Advanced High-Strength Steels”. In: *Welding Journal* (2008), p. 5.

[Ma+08] C. Ma et al. “Microstructure and fracture characteristics of spot-welded DP600 steel”. In: *Materials Science and Engineering A* (2008), p. 13.

[MY11] M. Ma and H. Yi. “Lightweight car body and application of high strength steels”. In: *Advanced Steels*. Springer, 2011, pp. 187–198.

[Mah07] P. Mahelle. “Caractérisation expérimentale et numérique du comportement d’assemblages soudés soumis à des sollicitations quasi-statiques et dynamiques”. PhD thesis. 2007.

[MD09] M. W. Mahoney and P. Drineas. “CUR matrix decompositions for improved data analysis”. In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702.

[MBP12] J. Mairal, F. Bach, and J. Ponce. "Task-Driven Dictionary Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.4 (2012), pp. 791–804.

[Mic16] A. Michel. "Pièces mécaniques soudées - Comportement élastique". In: *Techniques de l'ingénieur* (2016), p. 23.

[MH+04] M. A. Minnicino, D. A. Hopkins, et al. *Overview of reduction methods and their implementation into finite-element local-to-global techniques.* Tech. rep. Army Research Laboratory, 2004.

[Omo89] S. M. Omohundro. *Five balltree construction algorithms.* International Computer Science Institute Berkeley, 1989.

[PC95] K. Pal and D. L. Cronin. "Static and Dynamic Characteristics of Spot Welded Sheet Metal Beams". In: *Journal of Engineering for Industry* 117.3 (1995), pp. 316–322.

[Pal+05] M. Palmonella et al. "Finite element models of spot welds in structural dynamics: review and updating". In: *Computers & Structures* 83.8 (2005), pp. 648–661.

[Par+09] A. Parente et al. "Identification of low-dimensional manifolds in turbulent flames". In: *Proceedings of the Combustion Institute* 32.1 (2009), pp. 1579–1586.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[Pou+07] M. Pouranvari et al. "Effect of weld nugget size on overload failure mode of resistance spot welds". In: *Science and Technology of Welding and Joining* 12.3 (2007), pp. 217–225.

[Rei+19] A. Reille et al. "Incremental dynamic mode decomposition: A reduced-model learner operating at the low-data limit". In: *Comptes Rendus Mécanique* 347.11 (2019), pp. 780–792.

[Rei+21] A. Reille et al. "Learning data-driven reduced elastic and inelastic models of spot-welded patches". In: *Mechanics & Industry* 22 (2021), p. 32.

[Roe+16] D. Roettgen et al. “A comparison of reduced order modeling techniques used in dynamic substructuring”. In: *Dynamics of Coupled Structures, Volume 4*. Springer, 2016, pp. 511–528.

[Row00] S. T. Roweis. “Nonlinear Dimensionality Reduction by Locally Linear Embedding”. In: *Science* 290.5500 (2000), pp. 2323–2326.

[RSG95] A. Rupp, K. Störzel, and V. Grubisic. *Computer Aided Dimensioning of Spot-Welded Automotive Structures*. SAE Technical Paper 950711. SAE International, 1995.

[Ryc09] D. Ryckelynck. “Hyper-reduction of mechanical models involving internal variables”. In: *International Journal for Numerical Methods in Engineering* 77.1 (2009), pp. 75–89.

[Ryc+06] D. Ryckelynck et al. “On the a priori model reduction: Overview and recent developments”. In: *Archives of Computational Methods in Engineering* 13.1 (2006), pp. 91–128.

[SVV00] P. Salvini, F. Vivio, and V. Vullo. “Spot weld finite element for structural modelling”. In: *International Journal of Fatigue* 22 (2000), pp. 645–656.

[Sch10] P. J. Schmid. “Dynamic mode decomposition of numerical and experimental data”. In: *Journal of Fluid Mechanics* 656 (2010), pp. 5–28.

[SMB08] F. Seeger, G. Michel, and M. Blanquet. “Investigation of Spot Weld Behavior Using Detailed Modeling Technique”. In: *. LS* (2008), p. 10.

[See+05] F. Seeger et al. “An Investigation on Spot Weld Modelling for Crash Simulation with LS-DYNA”. In: *. LS* (2005), p. 13.

[She93] S. D. Sheppard. “Estimation of Fatigue Propagation Life in Resistance Spot Welds”. In: *Advances in Fatigue Lifetime Predictive Techniques: Second Volume* (1993).

[Sim+01] T. W. Simpson et al. “Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization”. In: *AIAA Journal* 39.12 (2001), pp. 2233–2241.

[Vim08] Srinivasan Vimalanathan. *An investigation on spot-weld modeling complexity for crash simulation*. 2008.

[Viv+02] F. Vivio et al. “Enforcing of an analytical solution of spot welds into finite element analysis for fatigue-life estimation”. In: *International Journal of Computer Applications in Technology* 15 (2002), pp. 218–229.

[WP05] D. -A. Wang and J. Pan. “A computational study of local stress intensity factor solutions for kinked cracks near spot welds in lap-shear specimens”. In: *International Journal of Solids and Structures* 42.24 (2005), pp. 6277–6298.

[Wan+06] J. Wang et al. “Simulation of Spot Weld Pullout by Modeling Failure Around Nugget”. In: SAE 2006 World Congress & Exhibition. 2006, pp. 2006–01–0532.

[Was+12] Kyle Washabaugh et al. “Nonlinear model reduction for CFD problems using local reduced-order bases”. In: *42nd AIAA Fluid Dynamics Conference and Exhibit*. 2012, p. 2686.

[Wem12] Klas Weman. *Welding processes handbook*. 2nd ed. Oxford ; Philadelphia: Woodhead Pub, 2012. 270 pp.

[Wil74] E. L. Wilson. “The static condensation algorithm”. In: *International Journal for Numerical Methods in Engineering* 8.1 (1974), pp. 198–203.

[Wu18] L. Wu. “Model order reduction and substructuring methods for nonlinear structural dynamics”. PhD thesis. 2018.

[XD04] S. Xu and X. Deng. “An evaluation of simplified finite element models for spot-welded joints”. In: *Finite Elements in Analysis and Design* 40.9 (2004), pp. 1175–1194.

[YGH09] J. Yvonnet, D. Gonzalez, and Q.-C. He. “Numerically explicit potentials for the homogenization of nonlinear elastic heterogeneous materials”. In: *Computer Methods in Applied Mechanics and Engineering* 198.33 (2009), pp. 2723–2737.

[ZT01] Y. Zhang and D. Taylor. “Optimization of spot-welded structures”. In: *Finite Elements in Analysis and Design* 37.12 (2001), pp. 1013–1022.

[ZS97] S. Zuniga and S. D. Sheppard. “Resistance Spot Weld Failure Loads and Modes in Overload Conditions”. In: *Fatigue and Fracture Mechanics: 27th Volume* (1997).

Appendix

Global resistance in weelding

The resistance R_{eq} models the resistance of the electrodes and clamped sheets. It can be measured by monitoring the current intensity and voltage at the electrodes terminals. Two types of resistances are involved in R_{eq}, the intrinsic resistances R_0, R_2, R_4 and the contact resistances R_1, R_3 and R_5 at the interfaces between the components (fig. 4.1). Even if they are polished, these interfaces present a roughness (a component of surface texture) of macroscopic and microscopic order. Thus, small distinct zones achieve contact. As electrical resistance varies inversely with the contact surface area, the contact resistances are higher compared to the intrinsic resistances when welding starts. Moreover, they increase as temperature rises in the matter. In particular, R_3 in the separating joint is higher than the two contact resistances between the electrode and the sheet R_1 and R_5. Hence, the temperature increase concentrates first in the separating joint as the cooling cycle of the electrodes conducts away most of the heat between the workpieces and the electrodes. Afterward, the contact at the interfaces improves because the mechanical pressure and the temperature rise soften the surface and smooth the roughness. Consequently, the contact resistances R_1, R_3 and R_5 drop rapidly while inside the metal the intrinsic resistance R_2 and R_4 rise with temperature. Therefore heating intensifies in the separating joint. A liquid weld nugget appears that grows up to the electrode's diameter. Subsequently, the current is shut down while pressure is maintained [Wem12].

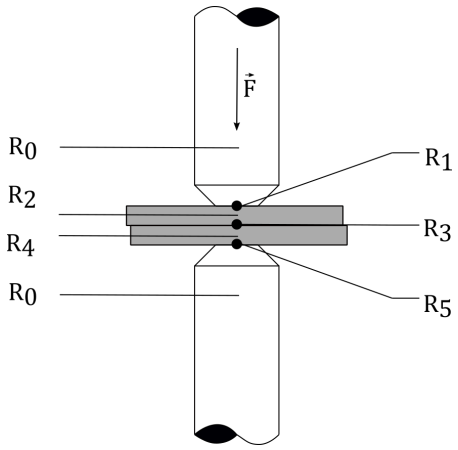


Figure 4.1 – The intrinsic resistance comprises R_0 in the electrodes, R_2 in the upper sheet and R_4 in the lower sheet. The contact resistances comprise R_1 between the upper electrode and the upper sheet, R_3 in the separating joint and R_5 between the lower electrode and the lower sheet.

Agathe REILLE

Réduction de modèle dans une intégration locale-globale, application aux points de soudure pour la simulation des crash-tests

Résumé

Calculer rapidement une simulation précise de crash-test reste un défi pour les constructeurs automobiles malgré la généralisation du calcul haute-performance et les progrès des méthodes numériques. En effet, des modèles à méso-échelle de petits composants sont intégrés dans le modèle global de la voiture pour obtenir une excellente précision sur l'ensemble mais l'introduction de ces modèles locaux augmente fortement le temps d'exécution du calcul. Les méthodes de réduction de modèle qui exploitent la dimension généralement faible de l'ensemble des sollicitations d'un phénomène physique et les corrélations entre celles-ci sont intéressantes pour améliorer la précision d'un modèle de crash-test à moindre coût. Cette thèse traite de la réduction des modèles méso-échelle des points de soudure dans le modèle de crash-test haute-fidélité d'ESI Group afin de diminuer le temps de calcul d'une simulation.

Nous proposons une nouvelle technique de réduction à partir de données et un schéma d'intégration introduisant un modèle réduit dans un modèle éléments finis. Ces méthodes permettent d'itérer précisément dans le domaine global au pas de temps de celui-ci.

Résumé en anglais

Performing a crash test simulation quickly and accurately remains a challenge for car manufacturers despite the generalization of high-performance computing and the progress in numerical methods. Indeed, engineers introduce mesoscale models of small components in the car model to obtain excellent prediction accuracy, but the integration of these local models increases computational time. Model order reduction methods that exploit the generally low dimensionality of the loading space of a physical phenomenon and the correlations in the loading are interesting to improve the accuracy of a crash test model at a low computational cost. In this thesis, we address the reduction of mesoscale spot weld models in ESI Group's high-fidelity crash test model to decrease the computational time of a simulation.

We propose a new data-driven reduction technique and an integration scheme to introduce this reduced model into a finite element model. We applied these techniques to reduce a mesoscale model of a spot weld. These methods allow to iterate precisely in the global domain at its time step.