



HAL
open science

Stratégie de modélisation multi-fidélité via une approche système incluant des métamodèles basés sur les entités NURBS

Bruno Vuillod

► **To cite this version:**

Bruno Vuillod. Stratégie de modélisation multi-fidélité via une approche système incluant des métamodèles basés sur les entités NURBS. Modélisation et simulation. HESAM Université, 2024. Français. NNT : 2024HESAE003 . tel-04439913

HAL Id: tel-04439913

<https://pastel.hal.science/tel-04439913>

Submitted on 5 Feb 2024

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

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

**ÉCOLE DOCTORALE SCIENCES DES MÉTIERS DE
L'INGÉNIEUR [CEA / Institut de Mécanique et d'Ingénierie de
Bordeaux (I2M) – Campus de Bordeaux]**

THÈSE

présentée par : **Bruno VUILLOD**

soutenue le : **10 janvier 2024**

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

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

Spécialité : **Sciences et Métiers de l'Ingénieur**

Stratégie de modélisation multi-fidélité via une approche système incluant des métamodèles basés sur les entités NURBS

THÈSE dirigée par :
M. MONTEMURRO Marco

co-dirigée par :
M. HALLO Ludovic

encadrée par
M. PANETTIERI Enrico

Jury

M. Amine AMMAR, Professeur des universités, LAMPA, ENSAM Angers,
M. Marco MONTEMURRO, Professeur des universités, I2M, ENSAM Bordeaux,
M. Ludovic HALLO, Directeur de recherche, CEA,
M. Enrico PANETTIERI, Maître de conférences, I2M, ENSAM Bordeaux,
Mme. Angela VINCENTI, Professeure des universités, Institut d'Alembert, UPMC,
M. Ludovic CHAMOIN, Professeur des universités, LMT, ENS Paris Saclay,
M. Tommaso TADDEI, Chargé de recherche, INRIA, IMB,

Président
Directeur de thèse
Directeur de thèse
Co-encadrant
Rapporteuse
Rapporteur
Examineur

Stratégie de modélisation multi-fidélité via une approche système incluant des métamodèles basés sur les entités NURBS

Thèse présentée par : Bruno VUILLOD
sous la direction de : M. Marco MONTEMURRO
co-dirigée par : M. Ludovic HALLO
encadrée par : M. Enrico PANETTIERI

Table des matières

Remerciements	vii
Acronymes	x
Introduction	3
1 État de l'art	11
1.1 Introduction	13
1.2 L'approche <i>Model-Based System Engineering</i>	14
1.2.1 Historique et présentation de l'approche <i>System Engineering</i>	14
1.2.2 L'approche Model-Based System Engineering	17
1.2.3 Le langage Modelica	19
1.2.4 Communication inter-logiciel	21
1.3 Analyse de sensibilité	24
1.3.1 Principe et historique de l'analyse de sensibilité	24
1.3.2 Analyse de sensibilité globale	25
1.4 Méthodes de métamodélisation	30
1.4.1 Méthodes de réduction d'ordre	30
1.4.2 Méthodes d'interpolation et d'approximation	37
1.4.3 Caractéristiques	46
1.5 Conclusion et stratégie complète	46
2 Identification des sous-systèmes critiques et influents	51
2.1 Introduction au chapitre	53
2.2 Analyse de sensibilité	53
2.2.1 Indices de Sobol et analyse de la variance	53
2.2.2 Indices de Shapley	58
2.3 Problème analytique de référence : la fonction d'Ishigami	58
2.4 Problèmes non-linéaires dynamiques - Approche <i>Model-Based System Engineering</i> (MBSE)	62
2.4.1 L'amortisseur de chocs	62
2.4.2 Oscillateur à frottement sec	68
2.5 Analyse de sensibilité globale sur des modèles Éléments Finis (EF)	72

2.5.1	Problème 1 : Analyse statique d'une structure treillis	72
2.5.2	Problème 2 : Analyse de flambement par valeurs propres d'un pan- neau raidi en composite mono-raisseur.	74
2.6	Conclusion du chapitre	79
3	Développement d'un métamodèle basé sur les entités Non-Uniform Ra- tional Basis-Spline	83
3.1	Introduction	85
3.2	Généralités sur les hypersurfaces <i>Non-Uniform Rational Basis-Spline</i> (NURBS)	86
3.3	Génération du métamodèle basé sur les entités <i>Non-Uniform Rational Basis- Spline</i> (NURBS) comme solution d'un problème d'optimisation	88
3.3.1	Description du problème	88
3.3.2	Initialisation du problème	89
3.3.3	Énoncé du problème d'optimisation	91
3.4	Mise en place du processus d'optimisation	95
3.4.1	Résolution du <i>Constrained Non-Linear Programming Problem</i> (CNLPP) pour l'évaluation des coordonnées des <i>Control Points</i> (CPs)	95
3.4.2	Optimisation numérique	101
3.5	Application du métamodèle sur des cas analytiques représentatifs	110
3.5.1	Influence du lissage sur les points de contrôle	111
3.5.2	Influence de l'optimisation des vecteurs nodaux et <i>overfitting</i>	113
3.5.3	Influence de l'optimisation du poids	115
3.6	Conclusion	117
4	Applications de la stratégie de métamodélisation basée sur les entités <i>Non-Uniform Rational Basis-Spline</i> (NURBS)	121
4.1	Introduction	123
4.2	Cas analytiques physiques non-linéaires	124
4.2.1	Équation de Poisson	124
4.2.2	Système oscillant non-linéaire	127
4.2.3	Système oscillant non-linéaire : évolution temporelle	129
4.2.4	Propagation d'une onde à la surface d'un fluide	132
4.3	Problèmes mécaniques nécessitant une résolution numérique	136
4.3.1	Plaque pleine	136
4.3.2	Plaque percée	138
4.3.3	Panneau raidi en flambement	142
4.4	Conclusion du chapitre	148
5	Application de la stratégie de modélisation multifidélité à un problème	

thermomécanique	151
5.1 Introduction au chapitre	153
5.2 Première modélisation <i>Model-Based System Engineering</i> (MBSE) et définition des objectifs	154
5.2.1 Analyse <i>Model-Based System Engineering</i> (MBSE) du Système Embarqué à Température Contrôlée-1	154
5.2.2 Analyse <i>Model-Based System Engineering</i> (MBSE) Système Embarqué à Température Contrôlée-2	158
5.3 Application de l'Analyse de Sensibilité Globale (ASG) pour la mise en évidence des sous-systèmes influents	161
5.3.1 Formulation du problème	162
5.3.2 Résultats numériques	163
5.4 Génération du métamodèle basé sur les hypersurfaces <i>Non-Uniform Rational Basis-Spline</i> (NURBS)	164
5.4.1 Zones critiques et modèle Éléments Finis (EF)	164
5.4.2 Génération du métamodèle comme solution d'un problème d'optimisation	167
5.4.3 Précision du métamodèle	169
5.5 Stratégie hybride haute-fidélité	171
5.5.1 Construction du modèle hybride haute-fidélité du Système Embarqué à Température Contrôlée (SETC)-3	171
5.5.2 Initialisation du modèle hybride haute-fidélité	171
5.5.3 Résultats du modèle hybride haute-fidélité Système Embarqué à Température Contrôlée (SETC)-3	172
5.6 Conclusion	175
Conclusion	179
A Annexe - Introduction générale	185
A.1 Valorisation de la Communication entre <i>Model-Based System Engineering</i> (MBSE) et CATIA dans un Contexte Hydro-mécanique	185
B Annexe - Chapitre 1	187
B.1 Rappel sur le calcul tensoriel	187
B.1.1 Algèbre tensorielle en 2 dimensions	187
B.1.2 Opérations	188
B.2 La manipulation symbolique pour le langage Modelica	189
C Annexe - Chapitre 2	191
C.1 Code Python pour le calcul des indices de Shapley	191

D Annexe - Chapitre 4	193
D.1 Plaque percée : phénomène d' <i>overfitting</i>	193
E Annexe - Chapitre 5	195
E.1 Modèles analytiques thermiques	195
E.2 Sous-systèmes <i>Model-Based System Engineering</i> (MBSE) globaux	197
E.3 Illustration du système thermique <i>Model-Based System Engineering</i> (MBSE) complet pour le cas Système Embarqué à Température Contrôlée (SETC)-2	198
E.4 Étude de convergence	199
Publications	201
E.5 Article publié dans une revue internationale, ou en préparation	201
E.6 Article publié dans un <i>proceeding</i> de conférence internationale	201
Bibliographie	218

Remerciements

Cette période consacrée à ma thèse a été une expérience enrichissante, aussi bien sur le plan professionnel que personnel. Cependant, je tiens à souligner que cette réalisation n'aurait pas été possible sans l'apport précieux de certaines personnes, que je tiens sincèrement à remercier.

Je souhaite exprimer ma profonde reconnaissance envers mon co-directeur de thèse, M. Ludovic HALLO. Notre collaboration a débuté lors de mon stage de fin d'études au CEA, où il m'a généreusement proposé de poursuivre ce travail en tant que doctorant. J'adresse également mes remerciements à M. Marco MONTEMURRO, mon directeur de thèse, dont l'engagement envers l'excellence académique ainsi que sa rigueur ont été une source d'inspiration constante pour moi. Un merci particulier à M. Enrico PANETTIERI, mon encadrant de thèse, qui m'a permis de découvrir le monde de l'enseignement. J'adresse un remerciement au CEA-CESTA pour avoir financé ce projet et pour avoir eu confiance en sa réussite. Mes remerciements vont également à M. Ludovic CHAMOIN et Mme. Angela VINCENTI, qui ont accepté d'évaluer mon manuscrit, ainsi qu'à M. Amine AMMAR et M. Tommaso TADDEI, qui ont examiné attentivement mon travail. Leurs commentaires pertinents ont enrichi cette thèse de différentes perspectives, et leur intérêt manifesté à mon égard est à la fois gratifiant et flatteur.

Je souhaite exprimer ma gratitude envers mes collègues, devenus des amis au cours de ces trois années d'aventure. En effet, une thèse représente bien plus que trois années d'études ou de formation à la recherche ; elle incarne également une période riche en bons et mauvais moments, que j'ai eu le privilège de célébrer et surmonter aux côtés de ces compagnons de route.

Au début, j'ai été soutenu par l'environnement du CEA, et en particulier du bureau bleu, où de grandes idées ont été échangées (salut Florent, Lucas, Gabriel et Florian). Parallèlement, mon travail au laboratoire I2M, notamment au département IMC, m'a offert une atmosphère quotidienne agréable au côté de : Romain, Chloé, Henri, Aude, Elisabetta, Camille, Simon, Pierre, George, Horly, Tugce, Sonia, Carlos, Thomas et Alexandre. Les liens avec les départements Dumas et MPI se sont renforcés au fil du temps, donnant lieu à des week-ends de cohésion inter-département mémorables. Un merci particulier à Mathilde, avec qui j'ai partagé une collaboration de deux ans et sans qui cette thèse n'aurait pas été aussi riche. Car une thèse représente également une aventure qui permet de se découvrir soi-même et de repousser ses propres limites, lorsque le cadre le permet.

La conclusion de ces huit années d'études représente une étape significative, largement appuyée par le soutien constant de mes parents et de ma famille, que je tiens à remercier chaleureusement. Particulièrement, mes parents ont joué un rôle essentiel dès le début de cette aventure, en m'aidant à décrocher un stage de découverte au CNRS. C'est grâce à leur soutien que j'ai pu commencer à explorer le monde de la recherche. Sous la supervision de M. Nicolas DESTAINVILLE, ce stage a été bien plus qu'une simple initiation ; il m'a permis d'échanger avec des doctorants et d'obtenir des perspectives éclairées sur leurs projets et le doctorat en général.

Enfin, mes remerciements vont à Eilleen, qui m'a accompagnée durant ces trois années, trouvant toujours les mots justes pour me féliciter dans les bons moments mais également pour me remonter le moral dans les moments difficiles. Je te suis reconnaissant d'être là, pour ton amour qui me comble de bonheur. J'attends avec impatience le début de notre vie future.

Acronymes

- AG** Algorithme Génétique
- ANN** *Artificial Neural Network*
- ANOVA** *ANalysis Of VAriance*
- ASG** Analyse de Sensibilité Globale
- ASL** Analyse de Sensibilité Locale
- AS** Analyse de Sensibilité
- CAO** Conception Assistée par Ordinateur
- CEA** Commissariat à l'Énergie Atomique et aux énergies alternatives
- CNLPP** *Constrained Non-Linear Programming Problem*
- CP** *Control Point*
- CPs** *Control Points*
- CS** *Co-Simulation*
- DEM** *Discrete Element Method*
- EAD** Équations Algébro-Différentielles
- EDO** Équations Différentielles Ordinaires
- EF** Éléments Finis
- ESPRIT** *European Strategic Programme for Research and development in Information Technologies*
- FAST** *Fourier Amplitude Sensitivity Test*
- FMI** *Functional Mock-up Interface*
- FMU** *Functional Mock-up Unit*
- GC-MMA** *Globally Convergent Method of Moving Asymptotes*
- I2M** Institut de Mécanique et d'Ingénierie
- IC** Intervalle de Confiance
- IHM** Interface Homme Machine
- INCOSE** *International Council On Systems Engineering*
- IRT** Institut de Recherche Technologique

LATIN *L*Arge *T*ime *I*Ncrement *m*ethod
LHC *L*ATIN *H*yper*C*ube
MBSA *M*odel-*B*ased *S*afety *A*nalysis
MBSE *M*odel-*B*ased *S*ystem *E*ngineering
MCP Matériau à Changement de Phase
MC Monte-Carlo
MDO *M*ultidisciplinary *D*esign *O*ptimisation
ME *M*odel *E*xchange
MHMN Modélisation Hybride MBSE - NURBS
MIMO *M*ultiple-*I*nter-*M*ultiple-*O*utput
MISO *M*ultiple-*I*nter-*S*ingle-*O*utput
MSL *M*odelica *S*tandard *L*ibrary
NURBS *N*on-*U*niform *R*ational *B*asis-*S*pline
OAT *O*ne-*A*t-*a*-*T*ime
PDF *P*robability *D*ensity *F*unction
PEC Plans d'Expérience Classiques
PEO Plans d'Expérience Optimaux
PE Plan d'Expérience
PPF Plans Factoriels Fractionnaires
PGD *P*roper *G*eneralised *D*ecomposition
POD *P*roper *O*rthogonal *D*ecomposition
QMC Quasi Monte-Carlo
RBD *R*andom *B*alance *D*esign
SETC Système Embarqué à Température Contrôlée
SE *S*ystem *E*ngineering
SIMO *S*ingle-*I*nter-*M*ultiple-*O*utput
SLSQP *S*equential *L*east *S*quares *P*rogramming
SVD *S*ingular *V*alue *D*ecomposition
TP *T*arget *P*oint
TPs *T*arget *P*oints
VF Volumes Finis
B-spline *B*asis-*S*pline
B-splines *B*asis-*S*plines
Gemseo *G*eneric *E*ngine for *M*ultidisciplinary *S*cenarios, *E*xploration and *O*ptimisation
SysML *S*ystems *M*odeling *L*anguage

Introduction

Dans un cycle de conception, la maîtrise de procédés incluant divers types de mécanismes physiques et de niveaux de modélisation devient un atout incontournable. En particulier, cette maîtrise est primordiale dans les premières étapes du cycle, qui correspondent au pré-dimensionnement du nouveau produit : l'ensemble du comportement du système d'étude doit être couvert. Les modélisations menées doivent être à la fois rapides, peu coûteuses et leur mise en place efficace. Elles doivent assurer un niveau de description au juste besoin en fonction de la zone du système d'étude considérée pour limiter le besoin en ressources de calcul. Cette approche répond aux besoins du bureau d'études du Commissariat à l'Énergie Atomique et aux énergies alternatives (CEA), où l'efficacité et la précision sont essentielles.

Cependant, la gestion de cette modélisation multiphysique et multifidélité présente un défi majeur. Pour reproduire des comportements complexes avec un minimum de ressources, une solution est l'emploi de métamodèles. Un métamodèle est une représentation simplifiée d'un modèle ou d'un système, généralement utilisé pour réduire leur complexité de mise en œuvre, accélérer les calculs et faciliter l'analyse. Ces métamodèles sont des modèles mathématiques légers, alimentés et calibrés par des bases de données construites à partir de résultats expérimentaux ou numériques, ces derniers issus de modèles haute-fidélité très gourmands en termes de ressources et temps de calcul. Ils sont conçus pour prédire avec précision le comportement du modèle d'origine tout en réduisant la charge de calcul.

Les avantages des métamodèles résident dans leur capacité à accélérer les simulations, à rendre les modèles plus accessibles aux ingénieurs et à réduire la demande de ressources informatiques. Ils sont intéressants dans diverses situations, notamment lorsque l'on ne dispose que de résultats expérimentaux sans loi de comportement, ou que des analyses statistiques ou des optimisations du modèle qui exigent de nombreuses évaluations rapides doivent être réalisées. Ces métamodèles jouent un rôle crucial dans la modélisation multifidélité, permettant de représenter un modèle complexe avec une précision ajustable, tout en maintenant la légèreté de la description mathématique.

Dans ce travail de recherche, la stratégie de métamodélisation est couplée à une approche *Model-Based System Engineering* (MBSE) pour élaborer une modélisation multifidélité de systèmes complexes. Cette approche innovante vise à rassembler dans un même modèle, des modèles basse-fidélité basés sur des descriptions analytiques d'Équa-

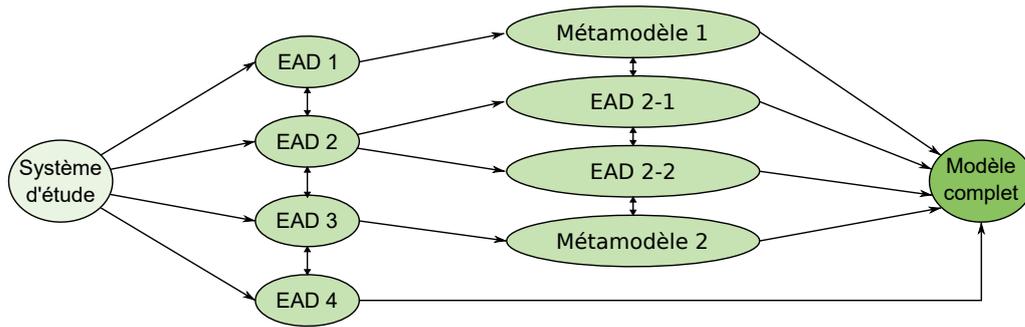


FIGURE 1 – Illustration de la démarche multifidélité.

tions Algèbre-Différentielles (EAD) et des modèles haute-fidélité peu coûteux, permis par un métamodèle. La démarche est illustrée dans la Fig. 1 et consiste à développer un premier modèle MBSE global avant de remplacer certains sous-systèmes analytiques par un métamodèle pour lui apporter le niveau de précision requis.

La finalité de cette stratégie est de fournir au bureau d'études du CEA, un modèle global de pré-dimensionnement prêt à l'emploi, qui peut être enrichi selon le besoin par l'incorporation de sous-systèmes basés sur des lois de comportement analytiques ou des calculs haute-fidélité apportés par le biais de métamodèles.

Le métamodèle développé dans le cadre de ce travail se base sur les hypersurfaces *Non-Uniform Rational Basis-Spline* (NURBS). Il s'agit d'une entité géométrique couramment utilisée dans la modélisation 3-D, la Conception Assistée par Ordinateur (CAO) et l'infographie. Elles sont particulièrement adaptées pour décrire des formes complexes et lisses, ce qui les rend indispensables dans l'industrie de la création visuelle, de la modélisation de surfaces et de la conception de produits. Ces entités sont définies par des *Control Points* (CPs), des vecteurs nodaux, et des poids. Contrairement à d'autres représentations, tels que les polygones, les NURBS offrent une précision élevée et une grande flexibilité pour créer des entités géométriques complexes multidimensionnelles, tout en conservant un contrôle local précis. Le terme *Non-Uniform Rational Basis-Spline* peut être décomposé comme suit :

- ***Non-Uniform*** : Les vecteurs nodaux associés à une dimension d'entrée ne sont pas uniformes. Ils permettent de définir les ensembles de variations des polynômes de Bernstein [138] employés dans la formulation des NURBS. En les déplaçant, cela permet de concentrer la précision là où elle est nécessaire, notamment au niveau des non-linéarités ;
- ***Rational*** : Les poids influencent l'hypersurface en attribuant plus ou moins d'importance aux CPs auxquels ils sont associés ;
- ***Basis*** : Les fonctions de base, correspondant aux polynômes de Bernstein, sont utilisées pour décrire la forme. Elles sont définies de manière récursive.

Les NURBS offrent de nombreux avantages, et une grande maniabilité. Leur em-

ploi initial en tant qu'entité géométrique dans les logiciels de conception 3-D a évolué dans plusieurs disciplines comme l'optimisation topologique [38, 39, 149], l'optimisation de forme [17, 128], l'optimisation du champ d'anisotropie pour les composites à rigidité variable [126, 127], et la génération de métamodèles [9–11], objet de ce travail.

Pour mettre en place la stratégie Modélisation Hybride MBSE - NURBS (MHMN), deux verrous ont du être levés. Le premier, scientifique, correspond au développement du métamodèle basé sur les hypersurfaces NURBS comme solution d'un processus d'optimisation sous contraintes. Le second, plus technique, vise à inclure le métamodèle dans l'approche MBSE après avoir identifié les sous-systèmes clefs. Pour répondre à ces problématiques, trois axes de recherches principaux ont été explorés :

1. le premier, innovant et technique, correspond à l'étude des possibilités et de la viabilité de l'approche MBSE, pour la modélisation et le pré-dimensionnement de systèmes multiphysiques complexes. Cela a été l'objet de deux stages préliminaires à cette thèse ainsi que d'un troisième stage encadré durant ce travail. Les deux premiers stages avaient pour but de proposer des démonstrateurs MBSE pour évaluer la pertinence de l'approche et son fonctionnement sur des cas d'intérêt du CEA. Le troisième a permis d'étudier les possibilités de couplage d'un modèle MBSE avec un environnement de modélisation 3-D sur la plateforme 3D Experience¹. Un résumé de ce dernier se trouve en annexe A.1. Suite à ce travail, le bureau d'études a jugé pertinent de poursuivre sur cette voie peu développée au sein du CEA ;
2. le second, innovant et scientifique, correspond à l'emploi de méthodes d'Analyse de Sensibilité Globale (ASG) sur un modèle MBSE. L'objectif est d'étudier la pertinence et le processus de mise en place de ces stratégies mathématiques pour aider à mettre en évidence les sous-systèmes principaux du modèle global MBSE. Une fois mise en évidence, la méthode de modélisation de ces sous-systèmes clefs doit être approfondie avec des méthodes plus performantes qu'une approche MBSE classique limitée à des modèles analytiques d'ordre zéro ;
3. le troisième et dernier axe de recherche, novateur et scientifique, correspond au développement de la stratégie de métamodélisation basée sur des hypersurfaces NURBS, comme solution d'un problème d'optimisation sous contraintes. Ce travail propose une généralisation aux cas *Multiple-Input-Multiple-Output* (MIMO) des algorithmes employés pour la résolution des problèmes de minimisation aux moindres carrés de [9]. Un terme de lissage, exprimé par [18, 56] pour le cas des problèmes de *surface fitting*, est généralisé au cas des hypersurfaces dans cette thèse. Ce terme sera intégré à la formulation du problème d'optimisation afin de limiter le phénomène d'*overfitting* et de réduire l'influence du bruit sur la génération du métamodèle. En outre, le gradient de la fonction de coût par rapport aux diverses variables d'optimi-

1. <https://www.3ds.com/>

sation sera exprimé sous forme analytique pour s'affranchir des limites de la méthode aux différences finies rencontrées dans [9, 18, 38].

Bien que la génération du métamodèle basé sur les entités NURBS se pose comme la solution d'un problème d'optimisation, cette thèse se concentre sur le développement de la génération des hypersurfaces NURBS ainsi que sur la dérivation analytique et l'implémentation des différents gradients intervenant dans le processus. Les méthodes d'optimisation au gradient employées sont des méthodes déjà existantes et robustes de la littérature, implémentées dans des bibliothèques *open source* ou propres au laboratoire Institut de Mécanique et d'Ingénierie (I2M).

L'efficacité de la stratégie de métamodélisation proposée dans cette thèse a été testée sur plusieurs cas d'application tirés de la littérature ainsi que sur des problèmes non-linéaires nécessitant une résolution numérique via la méthode aux Éléments Finis (EF).

La stratégie finale de la méthode de modélisation multifidélité MHMN est appliquée à un problème multiphysique simplifié qui correspond à un scénario thermomécanique non-linéaire représentatif des problèmes du CEA. Ce scénario inclut la plupart des comportements thermiques élémentaires, y compris les changements de phase qui induisent des comportements mécaniques complexes. L'approche MHMN se concentre sur l'organisation globale du problème à une échelle macroscopique, c'est-à-dire au stade de pré-dimensionnement, tout en maintenant un niveau de précision adéquat pour tirer des conclusions significatives à partir des tendances observées dans les différents sous-systèmes.

La stratégie de modélisation pourra être appliquée à des cas plus complexes et enrichis, par l'emploi de différents types de métamodèles selon le problème lié au sous-système à traiter. Une modélisation 3-D des dispositifs étudiés pourra également être apportée au modèle hybride final via les avantages permis par la plateforme 3D Experience de Dassault System. L'idée finale est d'alimenter des modèles complexes par un développement numérique réalisé par des physiciens. En parallèle, le bureau d'études pourra employer ces différents blocs élémentaires pour modéliser le système au juste besoin, en fonction notamment de l'avancement dans un cycle de conception.

Organisation du manuscrit

L'organisation de ce manuscrit suit le schéma de la Fig. 2. Le chapitre 1 présente ce qu'est l'approche *System Engineering* (SE) puis MBSE avant de proposer un tour d'horizon des principales méthodes d'Analyse de Sensibilité (AS), leurs avantages et leurs inconvénients. Une description non-exhaustive des principales méthodes de métamodélisation est ensuite faite avec notamment la présentation des méthodes de *Proper Orthogonal Decomposition* (POD), *Proper Generalised Decomposition* (PGD), *Artificial Neural Net-*

work (ANN) et NURBS. Une comparaison basée sur la littérature sera faite, pour mettre en avant les atouts des entités NURBS dans un contexte de métamodélisation. Pour finir, la stratégie MHMN est présentée.

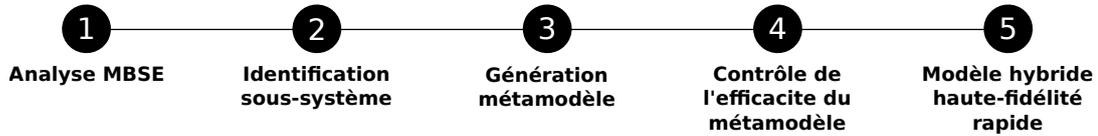


FIGURE 2 – Mise en place de la stratégie hybride de modélisation développée.

Le chapitre 2 porte sur l'application et la comparaison de méthodes d'ASG appliquées à des modèles non-linéaires en mécanique. La complexité des cas devient croissante pour étudier les limites de ce type de méthode. Dans ce contexte, le modèle d'un système est considéré comme étant une fonction de transfert qui exprime des sorties en fonction de variables d'entrée pour reproduire le comportement du système initial. Une analyse de sensibilité est un outil mathématique employé pour classer ces variables d'entrée en fonction de leur influence sur la (ou les) sortie d'intérêt. Ce chapitre 2 propose l'application des ASG basées sur les indices de Sobol et l'effet Shapley sur des systèmes modélisés par approche MBSE et EF.

Le chapitre 3 présente la stratégie de génération d'un métamodèle basé sur les hypersurfaces NURBS. Leur grande maniabilité en fait des métamodèles capables de s'adapter à une grande variété de problèmes MIMO, linéaires ou non. En contrepartie, cette souplesse a un défaut : la paramétrage des entités NURBS n'est pas une tâche anodine et obtenir un métamodèle précis et fiable peut se révéler être un vrai défi. Ainsi, la génération d'un métamodèle basé sur les entités NURBS se pose comme un problème d'optimisation sous contraintes dont les spécificités seront présentées et analysées dans le chapitre 3.

Dans le chapitre 4, nous mettons en pratique la méthode de création d'un métamodèle basé sur les hypersurfaces NURBS sur des problèmes d'intérêt. La première partie se concentre d'abord sur des cas physiques analytiques faiblement non-linéaires, puis sur des cas fortement non-linéaires. L'objectif est d'évaluer la capacité de la méthode à reproduire des comportements non-linéaires. Ensuite, nous appliquons le métamodèle à des problèmes de mécanique issus d'analyses haute-fidélité par EF. L'objectif est de démontrer la capacité du métamodèle à approximer des champs de déplacement et des comportements concrets. En outre, nous comparerons les temps de simulation entre les modèles haute-fidélité et le métamodèle.

Pour finir, le chapitre 5 propose l'application de la stratégie de MHMN développée tout au long de ce travail, sur un système thermomécanique non-linéaire. Ce dispositif correspond à un système embarqué simplifié qui comporte toutes les difficultés qu'un simple modèle MBSE ne peut résoudre : non-linéarités géométriques, non-linéarités matérielles, problème multiphysique, problème multifidélité.

Informations pour le lecteur :

Les acronymes ne seront définis qu'une seule fois dans le manuscrit mais sont rappelés dans la section *Acronymes*.

Les variables scalaires sont notées a , les vecteurs \mathbf{a} et les matrices et *array* \mathbf{A} .

1

État de l'art

Sommaire

1.1	Introduction	13
1.2	L'approche <i>Model-Based System Engineering</i>	14
1.2.1	Historique et présentation de l'approche <i>System Engineering</i>	14
1.2.2	L'approche Model-Based System Engineering	17
1.2.3	Le langage Modelica	19
1.2.4	Communication inter-logiciel	21
1.3	Analyse de sensibilité	24
1.3.1	Principe et historique de l'analyse de sensibilité	24
1.3.2	Analyse de sensibilité globale	25
1.3.2.1	Méthodes qualitatives - Le criblage	26
1.3.2.2	Méthodes quantitatives - Étude de la variance	28
1.4	Méthodes de métamodélisation	30
1.4.1	Méthodes de réduction d'ordre	30
1.4.1.1	Principe de la méthode Singular Value Decomposition	31
1.4.1.2	Principe de la méthode Proper Orthogonal Decomposition	34
1.4.1.3	Principe de la méthode Proper Generalised Decomposition	36
1.4.2	Méthodes d'interpolation et d'approximation	37
1.4.2.1	Les polynômes du chaos en métamodélisation	37
1.4.2.2	Régression déterministe - Régression linéaire	39
1.4.2.3	Régression stochastique - Processus gaussien et krigeage	40
1.4.2.4	Réseaux de neurones	41
1.4.2.5	Généralité sur les hypersurfaces <i>Non-Uniform Rational Basis-Spline</i>	43
1.4.3	Caractéristiques	46
1.5	Conclusion et stratégie complète	46

RÉSUMÉ DU CHAPITRE

Ce chapitre donne un aperçu de la modélisation système et propose un état de l'art des principaux sujets développés dans cette thèse. Une première partie place dans le contexte global du System Engineering l'approche Model-Based System Engineering (MBSE). Nous indiquerons comment cette méthode est utilisée dans ce travail, avec en perspective la possibilité de mettre en place des modélisations multifidélité, en fonction du taux d'avancement du projet. Nous montrerons également comment l'accroissement de précision de la modélisation MBSE peut s'appuyer sur l'utilisation de l'Analyse de Sensibilité (AS) et de méthodes de métamodélisation. Dans cette logique, la seconde partie du chapitre dresse une liste non exhaustive des méthodes d'AS au sens large, avant de se focaliser sur les méthodes d'Analyse de Sensibilité Globale (ASG). La troisième partie réalise un état de l'art des principales méthodes de métamodélisation avant de proposer un comparatif selon leurs caractéristiques intrinsèques. Pour conclure, une section présente dans son ensemble la stratégie de modélisation développée dans ce travail, qui associe les propriétés de l'approche MBSE, de l'ASG et de la méthode de métamodélisation proposée.

1.1 Introduction

Les méthodes de métamodélisation englobent un grand nombre de techniques permettant de réduire les coûts de calcul de systèmes d'étude. Toutes ces méthodes fonctionnent selon la même logique : un modèle reproduit le comportement du système étudié tout en minimisant son coût d'évaluation, que ce soit en termes de temps ou de ressources de calcul. Nous appellerons par la suite ce type de modèle « métamodèle ». Nous passerons en revue un certain nombre d'entre eux. En général, les métamodèles fonctionnent en deux phases : une partie d'apprentissage, dite *off-line*, où le métamodèle est calibré pour reproduire le comporte du modèle initial et une partie d'utilisation, dite *on-line*, durant laquelle le métamodèle est utilisé à la place du modèle initial. Ces métamodèles sont utiles dans des contextes tels que :

- la modélisation d'un système pour lequel on ne dispose que de résultats d'expérience et d'aucune loi de comportement ;
- la réduction du coût d'évaluation d'un modèle tout en conservant sa précision et sa robustesse ;
- l'analyse statistique du comportement du modèle et/ou son optimisation, nécessitant un grand nombre d'évaluations du modèle rapidement ;
- le dimensionnement d'un modèle, nécessitant une précision basse-fidélité dans un premier temps (modèle analytique d'ordre 0) puis haute-fidélité (modèle numérique d'ordre élevé).

Dans le cadre de ce travail, la métamodélisation est couplée à une approche MBSE. Cette dernière est employée pour modéliser de manière globale le système d'étude par une combinaison de sous-systèmes d'EAD (modèles physiques, modèles mathématiques, données tabulées, etc.). Ces derniers comportent souvent des hypothèses simplificatrices par rapport au comportement réel du sous-système. Le but est alors de les remplacer par des métamodèles avec une précision adaptable au juste besoin, sans pénaliser le coût de calcul du modèle global et sans introduire d'hypothèses simplificatrices (ou en limitant leur nombre). Le modèle résultant est multifidélité, avec une description globale gérée par la partie MBSE, et une description locale, gérée par l'inclusion d'un métamodèle et son niveau d'enrichissement.

Ce chapitre s'organise donc en trois parties avec en premier lieu la mise en contexte de cette stratégie innovante de modélisation. Il s'agit d'une méthode destinée à être utilisée en amont d'un cycle de conception, pour réaliser une première description multifidélité du système étudié, afin de compléter et d'approfondir l'étape de pré-dimensionnement. Ainsi, cette première section présente ce qu'est le SE, et plus particulièrement l'approche MBSE avec ses avantages et ses limites. Une seconde partie propose une liste non-exhaustive des principales méthodes d'AS avec notamment la distinction entre méthodes locales et glo-

bales. Ces dernières, employées dans ce travail, se différencient entre les méthodes qualitatives, représentées par la méthode de Morris, et les méthodes quantitatives, représentées par les indices de Sobol. Un état de l'art des méthodes de réduction de modèle est ensuite fait en troisième partie. Une distinction est à faire entre les méthodes d'interpolation et d'approximation et les méthodes de réduction d'ordre. La première définit un modèle mathématique, dont l'évaluation aux coordonnées de la base de données va fournir exactement la sortie correspondante. Par opposition, une méthode d'approximation va seulement approcher cette sortie, avec une erreur définie en amont. Ces deux approches vont reproduire le comportement de la base de données de travail mais pas le comportement intrinsèque du problème.

Pour ce qui est des méthodes de réduction d'ordre, elles vont permettre de réduire la complexité d'un problème par séparation de ses variables d'intérêt (aspect intrusif), tout en conservant le comportement intrinsèque du problème. Ce type de méthode approche ou interpole, selon le cas, le comportement du problème étudié. Les méthodes les plus communes sont la *Singular Value Decomposition* (SVD), la POD et la PGD. Les méthodes d'interpolation et/ou d'approximation non-intrusives évoquées plus loin sont les méthodes basées sur les polynômes du chaos, la régression linéaire simple et la régression linéaire faisant intervenir un terme gaussien, appelée aussi méthode de krigeage et les méthodes basées sur des réseaux de neurones. Pour finir la méthode de métamodélisation basée sur les entités NURBS est présentée. Ces entités sont une généralisation des courbes de Bézier et des *Basis-Splines* (B-splines), générées par la résolution d'un problème d'optimisation sous-contraintes et employées ensuite en tant que métamodèle. Une conclusion propose finalement une description générale de la stratégie développée dans ce travail, avec la mise en commun des 3 axes de recherches précédents.

1.2 L'approche *Model-Based System Engineering*

1.2.1 Historique et présentation de l'approche *System Engineering*

L'approche SE est un type d'approche holistique¹ et interdisciplinaire, devenue discipline à part entière, combinant ingénierie et management, ayant pour but de concevoir, intégrer et gérer des systèmes complexes tout au long de leur cycle de vie. Le terme est apparu pour la première fois en 1940 dans le laboratoire Bell Telephone et officiellement dans [157] en 1956. Le succès de ces modélisations, et ce de façon très précoce, est qu'elles sont adaptées aux besoins des physiciens, ingénieurs et aux bureaux d'études, étant donné qu'elles s'appuient sur la mise à plat des équations de bilan ou d'évolution, en faisant abstraction de tout problème de résolution numérique. La définition initiale a ensuite évo-

1. Selon le dictionnaire Larousse, adjectif tiré de *holisme* : mot utilisé par le Sud-Africain J. C. Smuts en 1926 pour indiquer la tendance de l'univers à construire des unités de complication croissante : matière inerte, matière vivante, matière vivante et pensante.

lué au cours des années, pour adapter la méthode à l'accroissement de la complexité des systèmes à résoudre. Cette évolution a donné lieu à deux approches complémentaires : l'approche *Multidisciplinary Design Optimisation* (MDO) au début des années 1970 et l'approche MBSE au début des années 1990 [190]. Officiellement, selon la société *International Council On Systems Engineering* (INCOSE)², l'approche MBSE correspond à l'application formalisée de la modélisation pour soutenir les exigences, la conception, l'analyse, la vérification et la validation des systèmes, dès la phase de conception et tout au long du développement et des phases ultérieures du cycle de vie. Quant à l'approche MDO, elle vient compléter l'approche MBSE par un travail d'analyse puis d'optimisation du modèle étudié. Demachy [47] propose, par exemple, une application couplée de l'approche MBSE avec l'approche *Model-Based Safety Analysis* (MBSA) faisant partie des méthodes MDO.

Un modèle SE peut être plus ou moins proche du comportement réel du système étudié, en fonction de la phase de conception et de l'objectif de l'utilisateur. Ce dernier peut à la fois s'intéresser au comportement global du modèle, de la même manière qu'à ses interactions avec son environnement de conception, de fabrication ou de fonctionnement. L'approche MBSE, quant à elle, consiste en une modélisation incluant au minimum une description structurelle et une description comportementale. Cela est réalisé par assemblage de blocs élémentaires, décrits par des lois de comportement sous forme de systèmes d'EAD. Il s'agit, en général, de la première modélisation, au niveau global, tenant compte des lois de comportement du système d'étude. Un état de l'art des principaux logiciels SE peut être trouvé dans [33], répondant chacun à un besoin particulier. Dans le cas présent, les plus pertinents sont :

- le logiciel Capela, développé par Thalès, basé sur la méthode Arcadia qui se divise en 4 niveaux : l'analyse opérationnelle, l'analyse système, l'analyse logique et l'analyse physique. Le besoin est exprimé et défini par les 2 premières analyses, puis une solution est construite et proposée par les 2 autres niveaux de la méthode. Ce logiciel couple l'utilisation des approches MDO et MBSE ;
- l'outil *Generic Engine for Multidisciplinary Scenarios, Exploration and Optimisation* (Gemseo)³, développé par l'Institut de Recherche Technologique (IRT) Saint Exupéry, est basé sur des codes Python permettant des analyses MDO. Il rassemble plusieurs outils numériques, mathématiques ou physiques parmi lesquels se trouvent les bibliothèques d'études statistiques OpenTURNS⁴ et Scipy⁵ ;
- *Systems Modeling Language* (SysML)⁶ est un langage permettant la description, l'analyse, la conception et la vérification des systèmes étudiés par emploi de gra-

2. <https://www.incose.org/>

3. <https://gemseo.readthedocs.io/en/stable/>

4. <http://openturns.github.io/openturns/latest/contents.html>

5. <https://scipy.org/>

6. <https://sysml.org/>

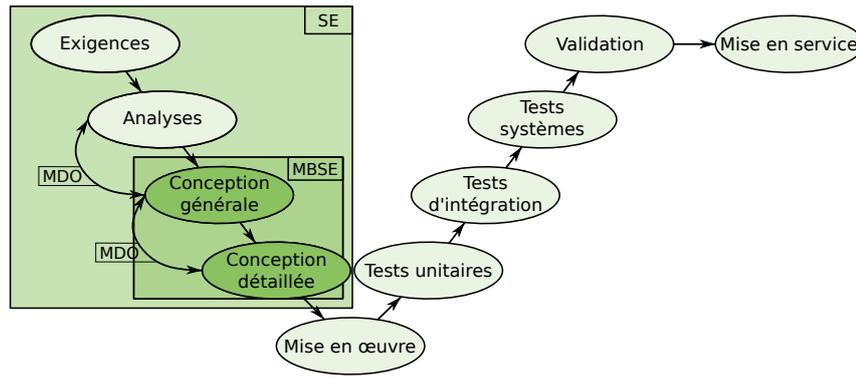


FIGURE 1.1 – Illustration du modèle de conception *V-and-V* avec positionnement des approches *System Engineering*, *Multidisciplinary Design Optimisation* et *Model-Based System Engineering* dans les différentes phases de ce cycle, inspiré de [148].

phiques. Ces derniers sont catégorisés en 3 ensembles : les diagrammes d'exigences, les diagrammes de fonctionnement et les diagrammes de structure ;

- le langage Modelica⁷, permettant une approche MBSE, s'apparente à Simulink (contrôle commande, société MathWorks⁸) ou à SPICE⁹ (électrocinétique), mais avec un aspect multiphysique. C'est un langage formel acausal orienté objet, destiné à la description dynamique de systèmes et de sous-systèmes mathématiques et physiques autonomes, pouvant être assemblés graphiquement entre eux pour modéliser un système.

La majeure partie des produits et services suivent un cycle de développement comparable au cycle de conception dit *V-and-V*, présenté dans [50, 141], appliqué dans [148] et illustré dans la Fig. 1.1 inspirée du même article. Ce cycle comprend un flux descendant durant lequel le besoin est exprimé jusqu'à la mise en œuvre de la solution en passant par des étapes intermédiaires d'expression du besoin, de son analyse et de sa conception. Un flux ascendant assure l'assemblage et l'application d'une série de tests en vue de la mise en service du produit.

La Fig. 1.1 met également en parallèle de ce cycle de conception, les phases d'emploi des différentes méthodes SE, MDO et MBSE présentées plus haut. On y retrouve ainsi la méthode SE, générale, englobant l'approche MBSE qui permet d'inclure les lois de comportements du système d'étude (phases de conception du cycle *V-and-V*). Le lien entre ces approches, i.e., les différentes phases du flux descendant du cycle *V-and-V*, est fait par l'application de méthodes MDO.

Initialement rassemblées en une seule étape, les phases de conception du cycle *V-and-V* ont été divisées par [57] en une étape générale et une étape détaillée. La conception générale va permettre de modéliser le cas d'étude par assemblage de sous-systèmes EAD, selon

7. <https://modelica.org/>

8. <https://fr.mathworks.com/>

9. <https://www.spice-space.org/>

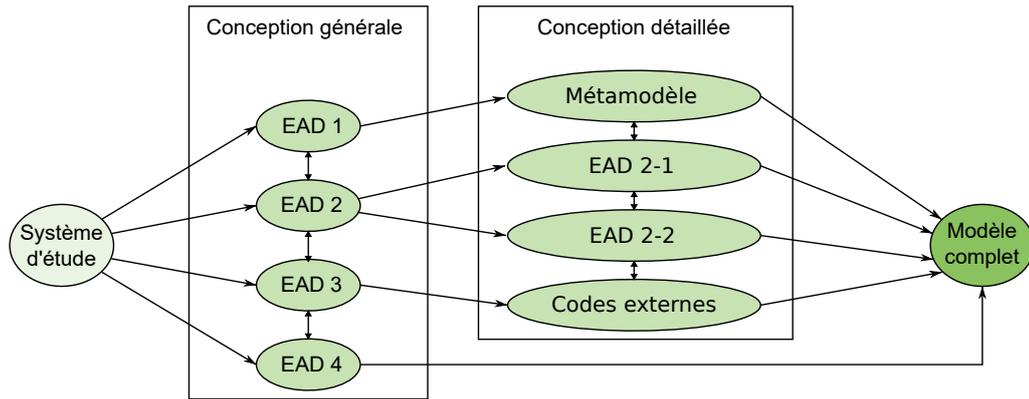


FIGURE 1.2 – Étape de *Conception générale* et de *Conception détaillée* du modèle *V-and-V*, avec des possibilités différentes pour accroître la précision d'un sous-système d'Équations Algébro-Différentielles.

l'approche MBSE classique. Dans un second temps, la conception détaillée va permettre d'approfondir, si nécessaire, les descriptions des sous-systèmes EAD critiques ou influents. Plusieurs solutions sont alors disponibles, illustrées par la Fig. 1.2 : l'emploi d'un modèle MBSE plus poussé, l'inclusion de codes externes spécifiques au sous-système ou encore l'inclusion de métamodèles, qui est la solution originale étudiée dans cette thèse.

Avant de réaliser un état de l'art sur les méthodes d'identification des sous-systèmes influents et sur les méthodes de métamodélisation, les sections suivantes proposent une étude bibliographique plus avancée de l'approche MBSE.

1.2.2 L'approche Model-Based System Engineering

L'approche MBSE permet une modélisation simplifiée mais rapide de systèmes plus ou moins complexes. Son principal avantage est de pouvoir se consacrer à l'écriture de modèles physiques, indépendamment de toute contrainte numérique. Cela répond bien à la demande d'ingénieurs ou de physiciens qui peuvent se concentrer sur l'établissement des équations du problème à résoudre plutôt qu'à leur résolution numérique. La résolution d'un problème par un langage classique (Python, Fortran, C, C++, etc.) et par un langage MBSE est illustrée respectivement dans les Fig. 1.3a et Fig. 1.3b. Un langage de programmation classique nécessite que l'utilisateur renseigne le système EAD global et instancie l'ordre de résolution et d'appel aux différentes fonctions requises pour la résolution du problème. Avec un langage MBSE, l'utilisateur peut davantage se concentrer sur la physique du problème. Il lui suffit de renseigner les différentes équations entrant en jeu dans le système d'étude, ainsi que leurs variables de communication, scalaires ou continues. Lors de la simulation, le système EAD global décrivant le modèle est généré automatiquement puis résolu par l'Interface Homme Machine (IHM).

Ce type de raisonnement et de fonctionnement est vraiment approprié à la modélisation de systèmes dynamiques. Cependant, en plus de manquer de précision compte tenu des

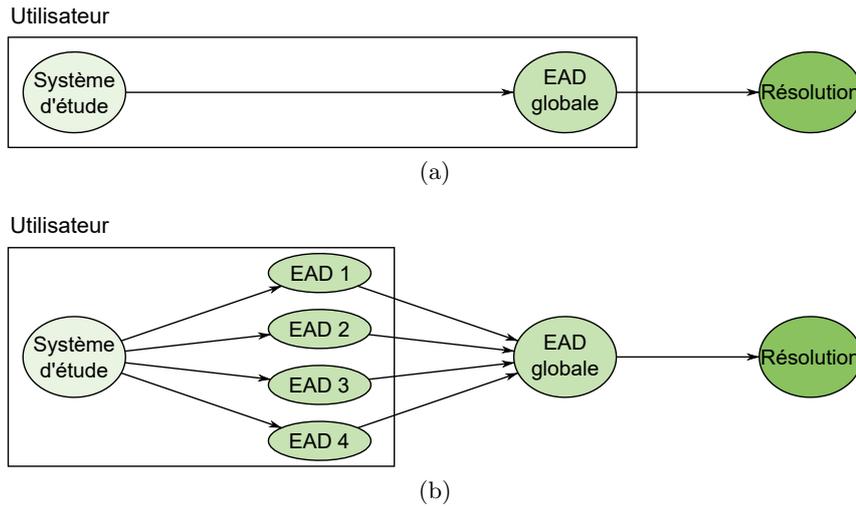


FIGURE 1.3 – Comparaison pour la résolution d’un problème quelconque par un langage classique (a) et une approche *Model-Based System Engineering* (b). Positionnement de l’utilisateur dans ces deux processus.

hypothèses réductrices souvent liées aux systèmes EAD, cette approche n’est pas adaptée à la résolution de systèmes spatiaux. En effet, pour des problèmes simples ou à but didactique, l’approche MBSE reste efficace et pratique d’emploi. A contrario, pour des problèmes plus poussés comme l’évolution spatio-temporelle de gradients de température nécessitant plusieurs milliers, voire plusieurs millions, de degrés de liberté, une approche par EF ou par Volumes Finis (VF) sera bien plus appropriée et plus systématique. Un exemple significatif est la résolution de l’équation de la chaleur sur une barre en une dimension avec un terme source :

$$\frac{\partial}{\partial t}T(x, t) + \alpha \frac{\partial^2}{\partial x^2}T(x, t) = \phi \quad (1.1)$$

avec α la diffusivité thermique dans le matériau considéré, $T(x, t)$ sa température à la position x au temps t et ϕ une source de chaleur. L’emploi de la méthode des EF nécessite, pour la résolution de l’Eq. (1.1), la construction d’une géométrie à une dimension, puis sa discrétisation via un partitionnement de la géométrie. Avec une approche MBSE, pour pouvoir obtenir la même discrétisation spatiale du gradient de température dans la phase transitoire, il est nécessaire de créer autant de sous-systèmes qu’il y a de mailles à prendre en compte avant la résolution. Un compromis serait d’employer la méthode *Discrete Element Method* (DEM), via le logiciel Granoo [6] par exemple. Elle permet de modéliser un milieu continu par assemblage d’éléments simples : des sphères et des poutres. Selon le cas, ces éléments peuvent avoir une réalité physique (lits de poudre [197]) ou pas (modélisation de milieux continus [5]). Cette méthode s’apparente à une approche MBSE avec une gestion efficace d’un grand nombre de sous-systèmes élémentaires identiques. Néanmoins, elle reste aussi coûteuse que la méthode des EF et aussi peu adaptée à l’inclusion de sous-systèmes multiphysiques dynamiques externes à la géométrie du problème (contrôle

commande par exemple). Bien qu'elles puissent tendre vers le même résultat dans certains cas, les définitions intrinsèques de ces méthodes les destinent à des emplois bien différents.

Les sous-systèmes d'EAD, qui composent le modèle global MBSE, comportent généralement des hypothèses réductrices du problème et ne sont pas adaptés à la résolution en espace. Pour conserver les avantages de l'approche MBSE tout en augmentant la précision de ses sous-systèmes critiques, cette thèse propose une alternative hybride et originale : l'inclusion de métamodèles externes, générés à partir de simulations numériques haute-fidélité, dans un modèle MBSE principal. Les problèmes traités dans ce travail étant physiques ou multiphysiques, le langage MBSE retenu est Modelica, plutôt que SPICE ou Simulink évoqués plus haut. En outre, la particularité du langage Modelica est sa capacité à résoudre des systèmes d'EAD avec une approche acausale combinée à la sémantique de Kirchhoff [186]. Un autre critère, commun à SPICE et Simulink, est que Modelica peut facilement communiquer avec des programmes et logiciels externes.

1.2.3 Le langage Modelica

Modelica [59] est un langage qui a été développé dans les années 1990 dans le cadre d'un projet inclus dans l'*European Strategic Programme for Research and development in Information Technologies (ESPRIT)*¹⁰. L'objectif était de mettre au point un standard de langage informatique en tenant compte des spécificités des langages existants. Une proposition pour un langage universel et simple d'utilisation a été faite : Modelica. En 1999, la première version utilisable de ce code a été créée sous la direction de Hilding Elmqvist, repris depuis par Dassault System. Cela a conduit ensuite à la création du logiciel Dymola, basé sur le langage Modelica et inclus dans la 3D Experience [45], plateforme inter-logiciels de Dassault System. Dans le même temps, l'association à but non lucratif *Modelica Association* a été créée. Aujourd'hui, elle continue à assurer les évolutions du langage et des compilateurs.

Le langage libre et non propriétaire Modelica est un langage formel orienté objet, destiné à la description dynamique de systèmes mathématiques. Dans le contexte de la modélisation physique, il décrit le comportement dynamique de systèmes complexes, pouvant impliquer plusieurs domaines de la physique. Son originalité réside dans le fait qu'il permet une description hiérarchique et acausale du système, propriétés intéressantes pour une approche MBSE. Les principales caractéristiques de Modelica sont explicitées ci-dessous et des exemples significatifs d'application peuvent être trouvés dans [26, 85, 137, 142, 179] :

- Modelica peut traiter des problèmes multiphysiques pouvant impliquer des phénomènes thermiques, mécaniques, hydrauliques ou encore biologiques, tant qu'ils sont

10. <https://cordis.europa.eu/programme/id/FP2-ESPRIT-2>

décrits par des modèles d'évolution temporelle provenant de systèmes d'EAD ou même de données tabulées ;

- Modelica est aujourd'hui intégré dans de nombreux logiciels, qu'il s'agisse de codes *open source* ou commerciaux, permettant un échange pratique de modèles. Les principaux logiciels basés sur ce langage sont Dymola (Dassault System¹¹), MapleSim (MapleSoft¹²) et OpenModelica (*Modelica Association*¹³);
- Modelica est un langage acausal : le système d'équations global n'est résolu que lorsque le code est compilé par manipulation symbolique. En d'autres termes, l'instanciation du code n'a pas d'importance sur sa résolution ;
- Modelica est destiné à être utilisé à travers une interface graphique où la modélisation se fait par assemblage de blocs élémentaires. Ils correspondent à des sous-systèmes à part entières, que l'utilisateur peut programmer ou trouver dans des bibliothèques open sources telles que la *Modelica Standard Library* (MSL), ThermoSysPro¹⁴, BuildSysPro¹⁵ ou encore ExternalData¹⁶.

L'exemple d'un système d'amortisseur classique modélisé par assemblages de sous-systèmes est présenté dans la Fig. 1.4. Les sous-systèmes en question sont une masse assemblée en série avec un ressort et un amortisseur en parallèle, eux-mêmes liés à un bâti, modélisé par un élément interdisant tout mouvement. L'ensemble est soumis à une charge modélisée par un sous-système de force. Le programme Modelica correspondant à ce modèle MBSE est généré automatiquement par le logiciel et disponible à l'édition pour l'utilisateur sous la forme suivante :

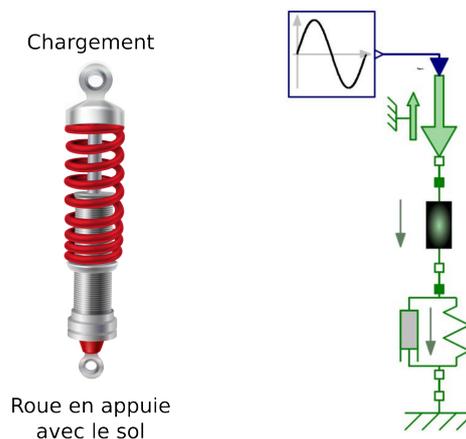


FIGURE 1.4 – Modélisation d'un système d'amortisseur par assemblage de sous-systèmes via l'interface graphique du logiciel Dymola, travaillant avec le langage Modelica.

11. <https://www.3ds.com/fr/>

12. <https://fr.maplesoft.com/index.aspx>

13. <https://modelica.org/>

14. <https://thermosyspro.com/install.html>

15. <https://github.com/EDF-Lab/BuildSysPro>

16. https://help.modelon.com/latest/tutorials/external_data_into_modelica_models/

```
1 model Amortisseur
2   Modelica.Mechanics.Translational.Components.Fixed fixed;
3   Modelica.Mechanics.Translational.Components.SpringDamper
4     springDamper;
5   Modelica.Mechanics.Translational.Components.Mass mass;
6   Modelica.Mechanics.Translational.Sources.Force force;
7   Modelica.Blocks.Sources.Sine sine;
8 equation
9   connect(sine.y, force.f);
10  connect(force.flange, mass.flange_a);
11  connect(mass.flange_b, springDamper.flange_b);
12  connect(springDamper.flange_a, fixed.flange);
13 end Amortisseur;
14
15 model Mass
16   parameter SI.Mass m(min=0, start=1) "Mass of the sliding mass";
17   SI.Velocity v(start=0) "Absolute velocity of component";
18   SI.Acceleration a(start=0) "Absolute acceleration of component";
19 equation
20   m*a = flange_a.f + flange_b.f;
21 end Mass;
22
23 model SpringDamper
24   SI.Force f_c "Spring force";
25   SI.Force f_d "Damping force";
26 equation
27   f_c = c*(s_rel - s_rel0);
28   f_d = d*v_rel;
29 end SpringDamper;
```

En résumé, le langage Modelica est capable de résoudre des systèmes complexes composés d'équations de bilan, d'équations différentielles et de potentiels, synthétisées dans un système global d'EAD. Cette résolution se fait par manipulation symbolique via des algorithmes spécifiques. Les algorithmes employés par le logiciel *open source* OpenModelica sont décrits en annexe B.2. Néanmoins, les logiciels commerciaux utilisent souvent la même résolution mais avec des algorithmes différents, souvent plus performants, qui ne sont pas accessibles dans la littérature. Pour plus de détails sur ce sujet, le lecteur est invité à se référer à [59].

1.2.4 Communication inter-logiciel

Un autre des atouts de Modelica, commun avec la plupart des langages MBSE, est sa facilité de communication avec d'autres langages et logiciels, ce qui est un des éléments

nécessaires à la mise en place de la modélisation multifidélité proposée dans cette thèse. Elle peut se faire sous différentes formes dont trois, illustrées par la Fig. 1.5, ont été explorées dans ces travaux :

- l’emploi d’un outil de gestion de logiciels tel que CMake¹⁷ ;
- la norme *Functional Mock-up Interface* (FMI) ;
- la plateforme 3D Expérience de Dassault System, regroupant plusieurs logiciels dont Dymola, qui est un logiciel basé sur le langage Modelica permettant l’approche MBSE.

Ce dernier moyen de communication ne sera pas décrit dans ce manuscrit mais a été l’objet d’un stage exploratoire encadré pendant cette thèse, résumé en annexe A. L’intérêt supplémentaire de cette approche est la communication entre le modèleur 3D CATIA et Dymola. Le premier permet de modéliser la géométrie de l’objet ainsi que les liens entre les différentes pièces qui le compose, tandis que le second, basé sur le langage Modelica, modélise les différentes physiques qu’il contient ainsi que les liaisons mécaniques d’un point de vue physique. Cette demande a été particulièrement appréciée par le bureau d’étude du CEA.

Le gestionnaire logiciel CMake permet de construire, tester et emballer des logiciels. Il est donc possible d’emballer un programme quelconque et de lui allouer une interface en C servant d’intermédiaire entre le logiciel d’import et le modèle global Modelica. Ce dernier est particulièrement bien adapté à ce moyen de communication car son fonctionnement se base sur du langage C. Cet aspect est détaillé en annexe B.2.

Le dernier moyen de communication du langage Modelica est la norme FMI. Son développement a été initié dans le cadre du projet européen MODELISAR en 2008¹⁸. L’objectif de cette norme est de permettre des connexions entre différents environnements de simulation et de modélisation afin de bénéficier de leurs fonctionnalités intrinsèques au sein d’un même projet. La dernière version utilisable de cette norme est présentée dans [19], mais la dernière mise à jour a été faite en juillet 2021 [68].

D’un point de vue numérique, la norme FMI définit une interface de communication, mise en œuvre au travers d’un fichier *Functional Mock-up Unit* (FMU). Une fois exporté sous ce format, le modèle ne dépend plus du logiciel d’export pour pouvoir fonctionner. D’un point de vue technique, un fichier FMU est assimilable à un fichier compressé contenant :

- un script Extensible Markup Language (XML) qui décrit la structure et les propriétés du modèle, utilisé pour communiquer avec le programme d’import ;
- le modèle lui-même, sous la forme d’un exécutable C ;

17. <https://cmake.org/>

18. <https://itea4.org/project/modelisar.html>

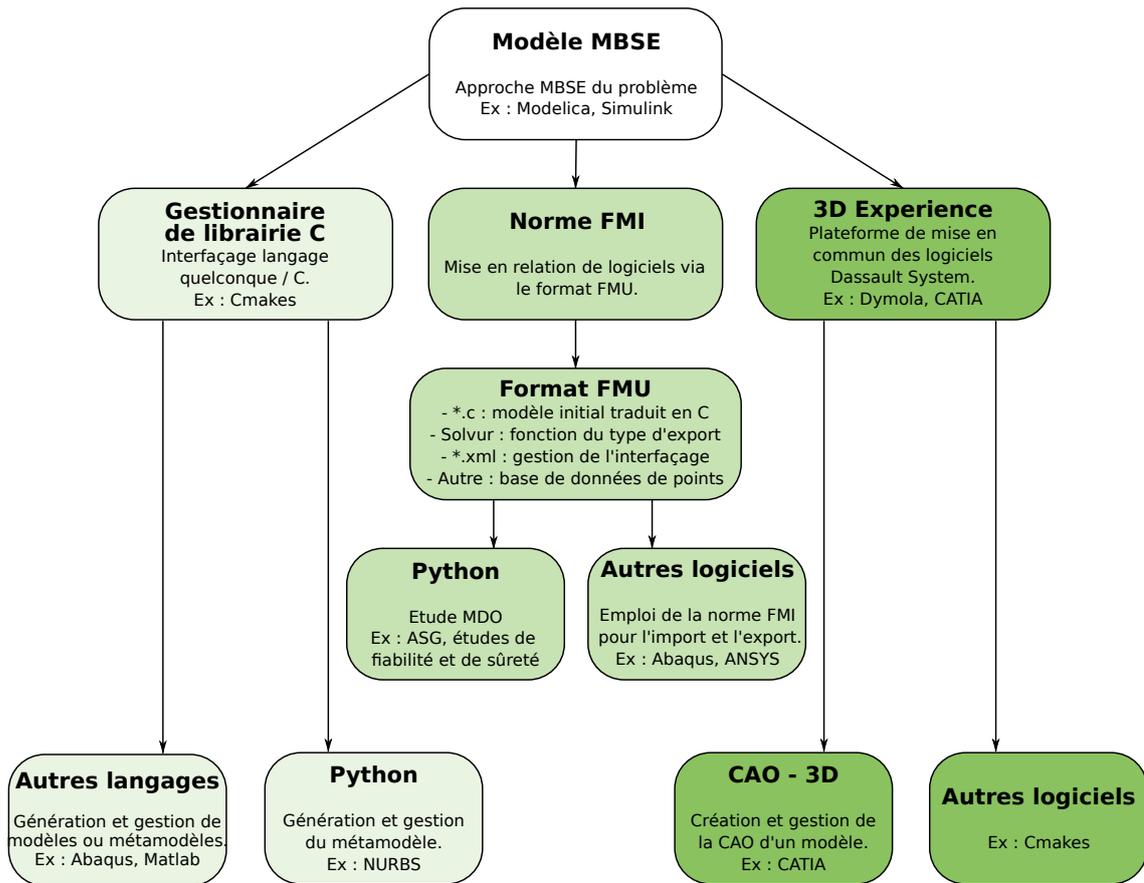


FIGURE 1.5 – Illustration de la norme *Functional Mock-up Interface* : un code principal permet l'appel et la mise en commun de différents codes. Ce code principal peut être mis en place via un logiciel (Modelica, Abaqus, Matlab), un script Python (PyFMI, FMPy) ou une interface graphique simple (Insight de MaplseSoft par exemple).

- les ressources et documents liés au modèle tels que les fichiers de points de données, les schémas d'icônes, etc. ;
- le solveur nécessaire à l'exécution du programme, selon le type d'export du FMU.

Lors de l'export d'un fichier au format FMU, un des paramètres primordial à définir est son type : *Co-Simulation* (CS) ou *Model Exchange* (ME)¹⁹. La différence est qu'en mode CS, le FMU comporte un solveur le rendant autonome, contrairement au mode ME résolu par le solveur du code d'import. Le mode d'export CS est particulièrement intéressant pour mener à bien des études statistiques. Sous ce format, le fichier FMU comporte à la fois le modèle sous la forme d'un exécutable ainsi qu'un solveur rendant les simulations plus rapides (jusqu'à un facteur 100 entre un modèle Modelica classique et son homologue FMU). Des exemples de l'utilisation de cette norme d'échange de modèles peuvent être trouvés dans [49, 76, 158, 173, 187]. Cette norme est particulièrement intéressante dans ces travaux, pour mener à bien les ASG requises dans la mise en place de la stratégie de

19. Ceci est vrai pour la version 2.0 de la norme FMI mais la nouvelle version, 3.0, permet un autre type d'export, encore en développement. Il ne sera donc pas évoqué ici

modélisation multifidélité.

1.3 Analyse de sensibilité

1.3.1 Principe et historique de l'analyse de sensibilité

Dans un contexte industriel où les produits sont de plus en plus complets, et nécessairement complexes, les modélisations deviennent coûteuses. Il peut donc être intéressant d'appliquer les efforts de calcul seulement sur des zones spécifiques du système d'étude. L'AS paramétrique peut aider à la décision dans ce choix. Il s'agit d'une méthode utilisée en amont du projet, qui permet d'identifier les paramètres influents d'un modèle. Pour rappel, un modèle est constitué d'une ou plusieurs variables d'entrées, d'une ou plusieurs sorties et d'une fonction de transfert traduisant le comportement d'un système numérique, mathématique, physique ou expérimental.

Plusieurs méthodes d'AS existent, applicables chacune dans un cadre qui lui est propre [89, 146, 154]. Néanmoins, elles fonctionnent toutes sur le même principe qui est de définir l'influence des paramètres d'étude en fonction du comportement des observables considérées. Un travail d'analyse est ensuite nécessaire pour répondre au besoin initial défini par l'utilisateur. Cela peut correspondre à une réduction du nombre de paramètres du modèle (en fixant les moins influents), ou encore à porter l'effort de modélisation sur un modèle spécifique du système global, pour en accroître la précision (second enjeu de cette thèse).

Parmi les méthodes d'AS, la première distinction est à faire entre les méthodes d'Analyse de Sensibilité Locale (ASL) et d'ASG. Les méthodes ASL permettent l'étude de la variance de la sortie d'un modèle en fonction de variations locales de ses entrées, autour de points nominaux [146, 154]. En d'autres termes, les paramètres d'entrée prennent des valeurs sur de petits intervalles de variation devant leur plage de variation complète. Cela est illustré dans la Fig. 1.6a. A l'inverse, comme le montre la Fig. 1.6b, les méthodes ASG prennent en compte les intervalles de variation complets de chaque paramètre considéré [89]. L'espace des sorties est ainsi parcouru dans sa globalité, et aucun cas particulier n'est écarté. C'est pour cette raison que seules des méthodes ASG sont employées dans ce travail.

Les méthodes ASG peuvent se diviser en quatre groupes [146] :

- les approches basées sur les dérivées partielles, qui sont des méthodes multi-locales, introduites par Morris [130] ;
- les approches qui consistent en une étude de variance, basée sur la décomposition de Hoeffding [81] et introduites par Sobol [164] ;
- les approches utilisant des variogrammes de variances, liant les approches basées sur les dérivées aux études de variances [144, 145] ;

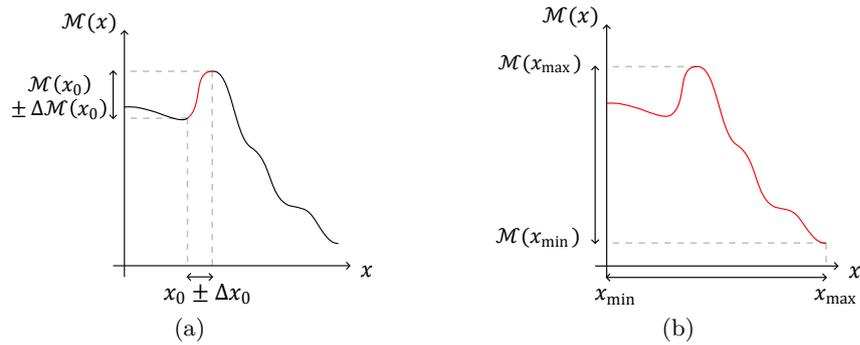


FIGURE 1.6 – Méthodes d’Analyse de Sensibilité Locale et d’Analyse de Sensibilité Globale illustrées sur un modèle $\mathbb{R}^1 \mapsto \mathbb{R}^1$. On peut observer sur le graphique (a) une variation locale de la sortie sur l’axe des ordonnées, due à une variation locale $\pm\Delta x_0$ du paramètre x_0 . Sur le graphique (b), on peut observer la variation complète du modèle en fonction de la variation du paramètre x sur tout son intervalle de variation. Cette figure est tirée de [72].

— les approches basées sur les régressions [97].

Plus largement, les méthodes ASG se distinguent entre les méthodes qualitatives et les méthodes quantitatives [89, 146, 154]. Les premières permettent une identification des paramètres peu influents sur la sortie considérée. Cela est intéressant pour réduire le nombre de paramètres d’entrée du modèle, lorsque celui-ci est important. Les principales méthodes qualitatives sont basées sur des Plan d’Expérience (PE) telle que la méthode de Morris [130]. D’un autre côté, les méthodes quantitatives permettent une hiérarchisation des paramètres d’entrée du modèle, par quantification de leurs influences. Elles se basent sur des échantillons de type Monte-Carlo (MC), ce qui les rend souvent plus coûteuses, mais plus précises. La méthode quantitative la plus commune est celle des indices de Sobol [164]. Au vu de ces définitions, la démarche généralement employée est la suivante : utiliser une méthode qualitative pour réduire le nombre de paramètres du modèle considéré (un nombre de paramètres est élevé s’il dépasse la dizaine / la vingtaine), puis appliquer une méthode quantitative pour hiérarchiser les paramètres influents restants. La répartition des principales méthodes qualitatives et quantitatives en fonction de la complexité du modèle traité et du nombre requis d’évaluations du modèle est illustrée dans la Fig. 1.7, inspirée de [88].

La section suivante propose de définir plus en détail la distinction à faire entre ces deux types de méthodes, notamment avec la présentation de la méthode de Morris et des indices de Sobol.

1.3.2 Analyse de sensibilité globale

Comme le montre la Fig. 1.7, les méthodes ASG se distinguent entre les méthodes basées sur du criblage, i.e., les méthodes qualitatives, et les méthodes basées sur la décomposition de la variance, i.e., les méthodes quantitatives. En outre, au sein même de

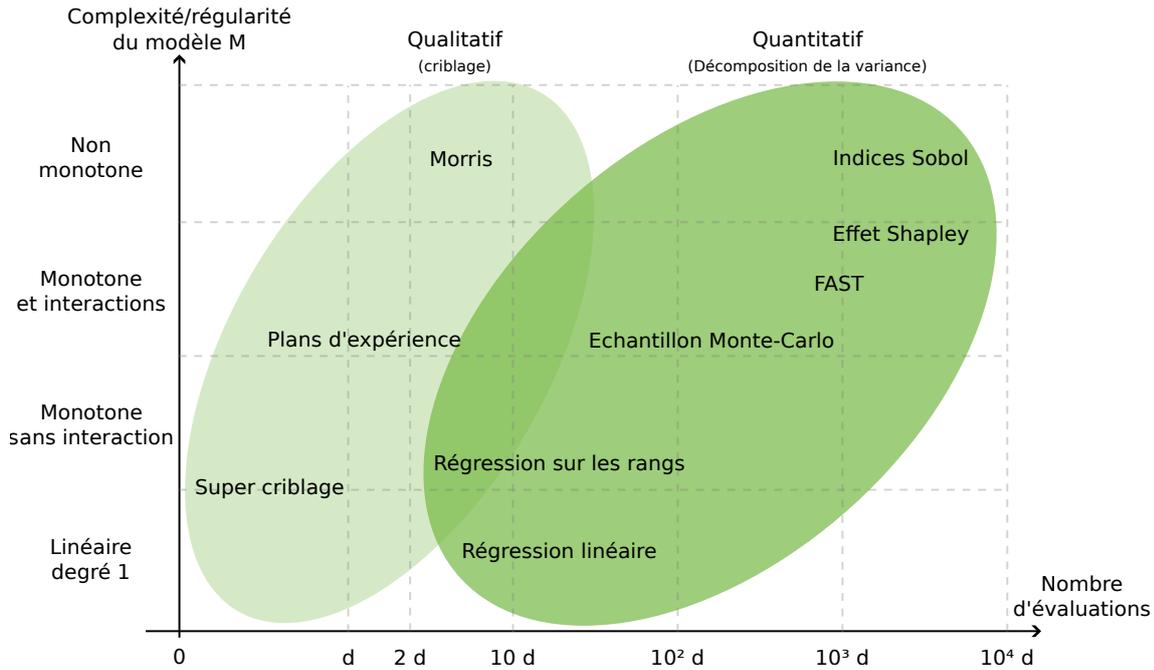


FIGURE 1.7 – Répartition des différentes méthodes d’Analyse de Sensibilité Globale en fonction de leurs spécificités suivant un schéma inspiré de [88].

ces deux groupes de méthodologies, plusieurs sous-méthodes sont à différencier. Dans cet état de l’art, et plus largement dans ce travail de thèse, les modèles soumis à une AS sont de type *Multiple-Input-Single-Output* (MISO) et comportent un petit nombre (inférieur à 20) de paramètres d’entrée indépendants, i.e., qui ne s’expriment pas en fonction d’autres paramètres étudiés dans l’ASG. Néanmoins, lorsque des paramètres sont corrélés, des solutions existent comme celles présentées dans [30, 31], qui revisitent des méthodes de Hooker [83] et Stone [172], tout comme les modèles MIMO [156, 196].

1.3.2.1 Méthodes qualitatives - Le criblage

Lorsqu’un modèle possède un grand nombre de variables d’entrée, l’objectif est souvent de le réduire pour diminuer le coût d’analyse du problème [37]. Plusieurs méthodes existent comme la théorie *active subspace* [37] ou les méthodes ASG qualitatives. Ces dernières se situent à une échelle globale du modèle et ont pour but de mettre en évidence les variables peu influentes. Elles sont ensuite fixées à leur valeur nominale et négligées dans la suite de l’étude. Ces méthodes fonctionnent généralement sur des PE qui nécessitent un nombre de simulations inférieur à la centaine, couplées à des analyses par criblage. La méthode de Morris est la plus connue de ces stratégies [130].

Les PE permettent de définir, en amont de projet, les jeux de variables des simulations les plus représentatives du comportement du modèle d’étude. Ils sont utiles pour réaliser des ASG efficaces mais également pour construire des métamodèles. Il en existe quatre types principaux [51, 70, 98] :

- les premiers, les Plans d'Expérience Classiques (PEC), vont chercher à recouvrir l'espace des entrées de manière optimale en fonction du modèle traité. Pour une fonction affine par exemple, seules les simulations aux bornes du domaine suffisent ;
- pour des modèles plus complexes, il faut utiliser des Plans d'Expérience Optimaux (PEO). Ils sont construits en considérant un critère de minimisation lié au modèle traité [8, 64] ;
- un autre type de PE sont ceux indépendants du modèle traité. L'objectif est de recouvrir au mieux l'espace des entrées à l'aide de différentes stratégies. La plus connue est celle de MC [122]. Néanmoins, la validité de cette théorie se base sur la loi des grands nombres, la rendant coûteuse à mettre en place pour obtenir un recouvrement de l'ensemble des possibles avec un niveau de confiance suffisant. Des algorithmes avec une plus grande vitesse de convergence ont alors été développés, comme celui de Sobol [163], celui de Halton [74] ou encore, plus simplement, le remplissage selon une grille uniforme. Ces méthodes sont caractérisées par un critère de discrédance et de remplissage d'espace (*space filling*) liés au nombre de tirages. Le premier quantifie la manière dont est rempli l'espace par rapport à une complétion uniforme (grille). Le second mesure la quantité de vide entre les points ;
- le dernier type de PE est celui des hybrides. Ils sont construits au fur et à mesure des simulations, en fonction des résultats fournis. L'espace des entrées est complété dans les zones où l'information est manquante, par exemple où l'erreur est maximale [51, 70].

Pour revenir aux méthodes ASG qualitatives, elles se basent sur un petit nombre de simulations définies par un PE couplé à un processus de criblage. Par définition, le criblage correspond à la sélection rapide de paramètres importants parmi un grand nombre de paramètres. Les méthodes classiques du criblage se basent sur des PEC type Plans Factoriels Fractionnaires (PFF) de résolution III ou IV²⁰. En d'autres termes, il s'agit de PEC réduits à l'aide de méthodes de sélection de cas représentatifs [70]. A titre d'illustration, en présence d'un modèle linéaire, les PEC vont faire varier ses coefficients sur leur plage de variation pour en déduire les zones instables et les variables influentes associées. La limite de ces modes d'AS se situe sur le type des modèles traités : s'ils sont trop fortement non-linéaires, non monotones ou comportent des paramètres corrélés, la méthode n'est pas applicable.

Pour contourner ces limites tout en restant dans le cadre des ASG qualitatives, il est préférable d'utiliser une autre méthode, comme celle de Morris [130]. Elle propose une estimation des effets élémentaires, i.e., de l'influence de chacun des paramètres, sans considérer leurs interactions. Pour ce faire, cette méthode se base sur la mise en place d'une

²⁰ La résolution est un raccourci permettant de quantifier le degré de confusion. Selon le type de résolution, les interactions entre les différentes variables sont considérés ou non [72]

succession d'études de type *One-At-a-Time* (OAT). Selon [131], il s'agit de faire varier une variable à la fois entre deux simulations successives en considérant N simulations. Les moyennes absolues μ^* des différentes variances associées aux variables sont calculées, tout comme les écart-types σ associés. Le diagramme des écarts-types σ en fonction des moyennes absolues μ^* est ensuite tracé. Ce diagramme, illustré dans la Fig. 1.8, permet alors de classer les variables du modèle selon leur influence et leur linéarité. Un exemple d'application est proposé par [62].

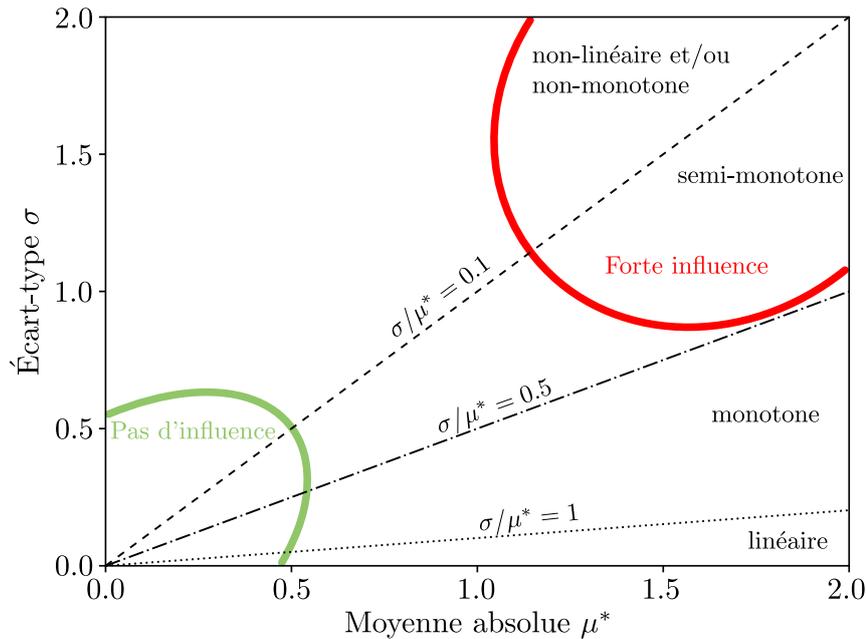


FIGURE 1.8 – Diagramme traçant la valeur des écart-types σ des effets élémentaires en fonction de leur valeur moyenne absolue μ^* . Identification des différentes variables de l'analyse selon leur influence et leur linéarité. Figure inspirée de [87].

1.3.2.2 Méthodes quantitatives - Étude de la variance

Comme illustré dans [62], une fois le nombre de variables réduit par une méthode qualitative, une méthode quantitative va compléter la première analyse en hiérarchisant précisément l'influence des variables d'entrée restantes. Ces méthodes se basant sur la décomposition de la variance de l'observable d'intérêt, il est nécessaire de recouvrir au mieux l'espace de variation de la sortie étudiée. La mise en place d'un PE a priori est donc nécessaire. Un tirage aléatoire selon la méthode MC est le plus commun mais n'est pas le plus efficace : l'emploi d'autres PE est souvent préférable, comme détaillé plus loin.

La décomposition de la variance utilisée dans les principales méthodes d'ASG se base sur la théorie de Hoeffding [81]. Brièvement, ces travaux permettent d'exprimer la variance de la sortie d'un modèle par sommation de variances conditionnelles de différents ordres inhérentes aux variables d'entrée du système d'étude : l'ordre 1 correspond à une variance

conditionnelle liée à une variables, l'ordre 2 à deux variables, etc. Ces variances conditionnelles sont ensuite normées par rapport à la variance globale, et utilisées pour calculer les indices de Sobol. Un indice de Sobol élémentaire, ou de premier ordre, correspond aux termes de premier ordre de la décomposition, divisés par la variance globale. Les indices de Sobol de second ordre correspondent aux termes de second ordre de la décomposition, divisés par la variance globale. Il en est de même pour les indices d'ordre supérieur. Les indices de Sobol totaux, sont un cas particulier. Ils correspondent à la somme de tous les indices précédents reliés à une variable d'entrée. Dans la pratique, seuls les indices élémentaires et totaux sont considérés. Les premiers caractérisent l'influence d'une variable seule sur la sortie, tandis que les seconds caractérisent l'influence d'une variable seule et de toutes ses interactions sur la sortie. Les indices d'ordre supérieur n'apportant pas d'informations plus importantes et étant coûteux à calculer, ils ne sont généralement pas évalués.

Néanmoins, une interaction entre deux variables (ou plus) est prise en compte aussi bien dans l'indice total de la première variable que dans l'indice total de la seconde variable. Pour déterminer la part d'influence exacte de chacune des variables, une solution est de considérer l'effet Shapley [159]. Il est couplé aux indices de Sobol de premier ordre, caractérisant ainsi l'influence complète d'une variable seule sur le modèle. Cela peut présenter un certains intérêt dans une ASG bien que leur calcul ne soit pas très robuste [180]. Ils seront donc calculés et comparés aux indices élémentaires et totaux pour les cas d'application considérés dans cette thèse (voir chapitre 2).

Comme évoqué, les indices de Sobol sont intéressants de par les résultats quantitatifs qu'ils fournissent mais couteux à déterminer sous leur forme initiale, elle-même basée sur un échantillonnage MC classique. Pour accélérer la convergence de ces indices, ou tout du moins pour réduire le coût de leur évaluation, différentes techniques ont été mises au point. Les principales sont les suivants :

- la re-formulation des expressions numériques des indices. En 2002, Saltelli [151] propose une stratégie d'évaluation de ces indices moitié moins coûteuse que précédemment. En 2010, Saltelli [152] a proposé plusieurs re-formulations des différents indices, certaines déjà existantes, d'autres proposées dans cet article [152] ;
- la re-formulation de la théorie. Dans [42], Cukier a proposé une nouvelle méthode, basée sur une analyse d'harmoniques déduites de transformées de Fourier, pour calculer les indices de Sobol : la méthode *Fourier Amplitude Sensitivity Test* (FAST). Cette stratégie a ensuite été approfondie par [155,175] puis encore améliorée par [67, 140,177] ;
- la réduction du coût propre des simulations. Pour ce faire, des métamodèles sont souvent employés comme les processus gaussiens tel que le krigeage [198]. D'un point de vue technique, des formats d'export particuliers peuvent aussi être utilisés comme le ".exe", ou le ".fmu", présenté à la section 1.2.4 et employé pour une ASG dans [179–

181].

L'ASG va être utilisée dans le cadre de cette thèse pour aider à la détermination des sous-systèmes influents d'un modèle global MBSE. Par la suite, les sous-systèmes cibles vont être modélisés par des calculs haute-fidélité et peu coûteux devant leur description analytique initiale. Cet état de l'art propose, dans la section suivante, un tour d'horizon non exhaustif des principales méthodes de métamodélisation, comparées entre elles qualitativement.

1.4 Méthodes de métamodélisation

Une fois le (ou les) sous-système critique du modèle global MBSE identifié, l'objectif va être d'accroître sa précision sans pénaliser les coûts de calcul du modèle global à l'aide d'un métamodèle ; l'objectif est d'obtenir le maximum de connaissance d'un système quelconque, à partir d'un minimum d'information. Plusieurs méthodes de métamodélisation existent en fonction du cas d'application et du besoin de l'utilisateur. Cet état de l'art propose de comparer certaines de ces méthodes selon les critères suivants :

- le coût de la partie *off-line*. Il s'agit de l'étape correspondant à la création du modèle réduit. Selon le cas, elle peut être *a priori* ou *a posteriori* et plus ou moins coûteuse ;
- sa polyvalence sur les modèles à traiter. Si celui-ci est linéaire monotone ou complètement non-linéaire, voire aléatoire, les méthodes à utiliser ne seront pas les mêmes ;
- la quantité de données nécessaires à la validité de la méthode ;
- l'intrusivité de la méthode, i.e., si cette dernière dépend du modèle traitée ou non et dans quelle mesure ;
- le coût de la partie *on-line* de la méthode, qui traduit son efficacité d'utilisation.

1.4.1 Méthodes de réduction d'ordre

Les méthodes de réduction de modèle sont les plus communes mais aussi les plus éprouvées [1]. Il s'agit des méthodes SVD, POD et PGD applicables sur des cas MIMO. La SVD est une méthode qui permet de trouver les valeurs et vecteurs propres du modèle d'étude là où la POD et la PGD vont s'appuyer sur ce principe pour résoudre un problème aux valeurs propres. En d'autres termes, un problème de dimension N va être réduit à un ou plusieurs sous-systèmes de dimension $n < N$. La solution globale est ensuite reconstruite par une somme de produits tensoriels de vecteurs propres (voir rappel sur le calcul tensoriel annexe B.1). Ces derniers correspondent tous respectivement à un mode de la solution : plus le nombre de modes considérés est grand, plus la solution est précise. Il est donc nécessaire de définir en parallèle un critère de convergence pour trouver un compromis sur le nombre de modes à considérer. En outre, ces méthodes se basent sur un algorithme

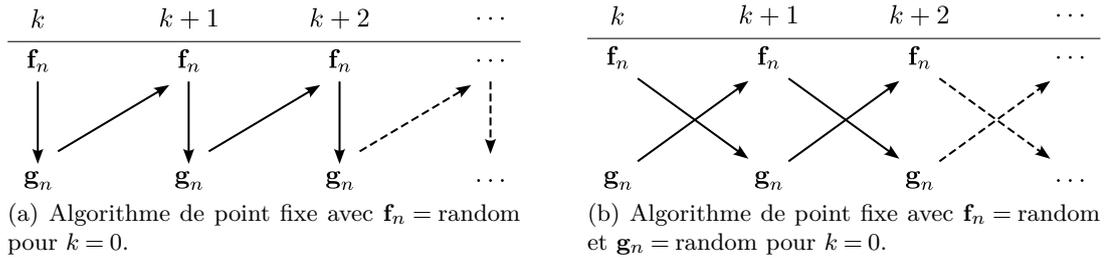


FIGURE 1.9 – Fonctionnement des algorithmes de point fixe. La méthode (a) où une seule fonction est initialisée par calcul stochastique et la méthode (b) où les deux fonctions (où plus selon le cas) sont évaluées par calculs stochastiques. La variable k correspond à la progression de la convergence de l'algorithme.

de point fixe pour pouvoir résoudre les sous-modèles. Il en existe de plusieurs types mais seulement deux seront décrits dans la section suivante traitant de la SVD.

1.4.1.1 Principe de la méthode Singular Value Decomposition

Cette méthode est connue pour sa stabilité ainsi que son efficacité dans l'orthogonalisation d'une base. Cela en fait un outil multi-disciplinaire, employé pour calculer les valeurs propres d'un problème [78]. Dans le cas de la réduction de modèle, cet outil peut être employé seul ou utilisé dans l'établissement des méthodes POD et PGD.

Classiquement, la SVD est une méthode qui permet de résoudre un problème aux valeurs propres. Pour le reste de la section, on considère un problème MISO dont la fonction de transfert est donnée par $\mathcal{M} : \mathcal{A} \subseteq \mathbb{R}^2 \rightarrow \mathcal{B} \subseteq \mathbb{R}$, avec \mathcal{A} le domaine de définition où les variables d'entrée prennent leur valeur et \mathcal{B} , le co-domaine où la sortie évolue. Une résolution SVD revient à exprimer la matrice \mathbf{M} dans $\sum_{n=0}^r \mathbf{f}_n \otimes \mathbf{g}_n = \mathbf{M}$, avec \mathbf{f}_n et \mathbf{g}_n les vecteurs propres liés à chacune des variables d'entrée et r correspondant au nombre de modes considérés. Ils sont évalués tour à tour via un algorithme de point fixe qui peut être défini de deux manières :

- la méthode (a) qui, à partir de \mathbf{f}_0 évaluée aléatoirement, détermine \mathbf{g}_0 puis \mathbf{f}_1 et \mathbf{g}_1 (ou l'inverse). Le processus est répété jusqu'à convergence et illustré dans la Fig. 1.9a ;
- la méthode (b) qui, à partir de \mathbf{f}_0 et \mathbf{g}_0 évaluées aléatoirement, détermine \mathbf{f}_1 et \mathbf{g}_1 . Le processus est ensuite répété jusqu'à convergence. L'inconvénient de cette méthode est l'introduction de deux erreurs au lieu d'une, liées aux deux tirages aléatoires. Cette méthode est illustrée dans la Fig. 1.9b.

En considérant la première méthode ainsi que l'opérateur produit scalaire $\langle \cdot, \cdot \rangle$, on détermine analytiquement les r différents termes comme suit :

$$\mathbf{M} = \mathbf{f}_n \otimes \mathbf{g}_n \iff \mathbf{g}_n = \frac{\mathbf{M}^T \mathbf{f}_n}{\langle \mathbf{f}_n, \mathbf{f}_n \rangle} \text{ et } \mathbf{f}_n = \frac{\mathbf{M} \mathbf{g}_n}{\langle \mathbf{g}_n, \mathbf{g}_n \rangle}. \quad (1.2)$$

Les calculs de \mathbf{f}_n et \mathbf{g}_n se succèdent jusqu'à l'atteinte d'un critère de convergence local,

ε , déterminé en amont, selon une norme \mathcal{L}_P notée $\|\cdot\|$:

$$\|\mathbf{f}_n^k \otimes \mathbf{g}_n^k - \mathbf{f}_n^{k+1} \otimes \mathbf{g}_n^{k+1}\|_P \leq \varepsilon. \quad (1.3)$$

Une fois le critère de convergence atteint, plusieurs options sont possibles :

- On conserve \mathbf{f}_n et \mathbf{g}_n tels quels ;
- On recalcule \mathbf{M} en prenant en compte un coefficient de normalisation α exprimé par :

$$\begin{cases} \alpha = \|\mathbf{f}_n\| \cdot \|\mathbf{g}_n\| \\ \mathbf{f}_n = \frac{\mathbf{f}_n}{\|\mathbf{f}_n\|} \\ \mathbf{g}_n = \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|} \end{cases} \implies \mathbf{M} = \alpha(\mathbf{f}_n \otimes \mathbf{g}_n). \quad (1.4)$$

- On recalcule $\mathbf{M} = \mathbf{f}_n \otimes \mathbf{g}_n$ en prenant en compte le même coefficient de normalisation α mais dans les expression de \mathbf{f}_n et \mathbf{g}_n directement :

$$\begin{aligned} \mathbf{f}_n &= \frac{\mathbf{f}_n}{\|\mathbf{f}_n\|} \sqrt{\alpha} = \mathbf{f}_n \sqrt{\frac{\mathbf{g}_n}{\mathbf{f}_n}}, \\ \mathbf{g}_n &= \frac{\mathbf{g}_n}{\|\mathbf{g}_n\|} \sqrt{\alpha} = \mathbf{g}_n \sqrt{\frac{\mathbf{f}_n}{\mathbf{g}_n}}. \end{aligned} \quad (1.5)$$

A l'itération 2, la solution globale \mathbf{M} est réduite d'un terme $\mathbf{f}_0 \otimes \mathbf{g}_0$ et on obtient :

$$\mathbf{M}' = \mathbf{M} - \mathbf{f}_0 \otimes \mathbf{g}_0. \quad (1.6)$$

Ainsi, de la même façon que précédemment, on tire aléatoirement \mathbf{g}_1 (ou \mathbf{f}_1) et on déduit :

$$\begin{aligned} (\mathbf{f}_1 \otimes \mathbf{g}_1)\mathbf{g}_1 &= \mathbf{M}\mathbf{g}_1 - \mathbf{M}'\mathbf{g}_1 \iff (\mathbf{f}_1 \otimes \mathbf{g}_1)\mathbf{g}_1 = \mathbf{M}\mathbf{g}_1 - (\mathbf{f}_0 \otimes \mathbf{g}_0)\mathbf{g}_1, \\ &\iff \mathbf{f}_1 \langle \mathbf{g}_1, \mathbf{g}_1 \rangle = \mathbf{M}\mathbf{g}_1 - \mathbf{f}_0 \langle \mathbf{g}_0, \mathbf{g}_1 \rangle, \\ &\iff \mathbf{f}_1 = \frac{\mathbf{M}\mathbf{g}_1 - \mathbf{f}_0 \langle \mathbf{g}_0, \mathbf{g}_1 \rangle}{\langle \mathbf{g}_1, \mathbf{g}_1 \rangle}. \end{aligned} \quad (1.7)$$

En généralisant le processus, il vient la suite ci-dessous :

$$\begin{cases} \mathbf{f}_{n+1} = \frac{\mathbf{M}\mathbf{g}_{n+1} - \sum_{i=0}^n \mathbf{f}_i \langle \mathbf{g}_i, \mathbf{g}_{i+1} \rangle}{\langle \mathbf{g}_{n+1}, \mathbf{g}_{n+1} \rangle}, \\ \mathbf{g}_{n+1} = \frac{\mathbf{M}^T \mathbf{f}_{n+1} - \sum_{i=0}^n \mathbf{g}_i \langle \mathbf{f}_i, \mathbf{f}_{i+1} \rangle}{\langle \mathbf{f}_{n+1}, \mathbf{f}_{n+1} \rangle}. \end{cases} \quad (1.8)$$

Finalement, on reconstruit le modèle global \mathbf{M} comprenant les r modes nécessaires à la convergence. Il est possible également que le modèle étudié soit déjà décomposé sous une forme du type : $\mathbf{M} = \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i$. Ainsi, en reprenant la formulation de l'Eq. (1.2) on en déduit que :

$$\begin{aligned}
\mathbf{f}_0 \otimes \mathbf{g}_0 &= \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i \iff \mathbf{f}_0^T (\mathbf{f}_0 \otimes \mathbf{g}_0) = \mathbf{f}_0^T \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i, \\
&\iff (\mathbf{f}_0^T \cdot \mathbf{f}_0) \mathbf{g}_0 = \mathbf{f}_0^T \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i, \\
&\iff \mathbf{g}_0 = \frac{\mathbf{f}_0^T \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i}{\mathbf{f}_0^T \cdot \mathbf{f}_0}, \\
&\iff \mathbf{g}_0 = \frac{\mathbf{f}_0^T \sum_{i=1}^r \mathbf{f}_i \otimes \mathbf{g}_i}{\langle \mathbf{f}_0, \mathbf{f}_0 \rangle}, \\
&\iff \mathbf{g}_0 = \frac{\sum_{i=1}^r \mathbf{g}_i \langle \mathbf{f}_i, \mathbf{f}_0 \rangle}{\langle \mathbf{f}_0, \mathbf{f}_0 \rangle}.
\end{aligned} \tag{1.9}$$

De la même manière on déduit que :

$$\mathbf{f}_0 = \frac{\sum_{i=1}^r \mathbf{f}_i \langle \mathbf{g}_i, \mathbf{g}_0 \rangle}{\langle \mathbf{g}_0, \mathbf{g}_0 \rangle}. \tag{1.10}$$

Ainsi, de manière générale, un problème dont la décomposition de variables est déjà connue s'exprime comme :

$$\begin{cases}
\mathbf{f}_{n+1} = \frac{\sum_{i=1}^r \mathbf{f}_i \langle \mathbf{g}_i, \mathbf{g}_{n+1} \rangle - \sum_{i=1}^r \mathbf{f}_i \langle \mathbf{g}_i, \mathbf{g}_{n+1} \rangle}{\langle \mathbf{g}_{n+1}, \mathbf{g}_{n+1} \rangle}, \\
\mathbf{g}_{n+1} = \frac{\sum_{i=1}^r \mathbf{g}_i \langle \mathbf{f}_i, \mathbf{f}_{n+1} \rangle - \sum_{i=1}^r \mathbf{g}_i \langle \mathbf{f}_i, \mathbf{f}_{n+1} \rangle}{\langle \mathbf{f}_{n+1}, \mathbf{f}_{n+1} \rangle}.
\end{cases} \tag{1.11}$$

De nouveau, deux critères de convergence sont à définir, l'un local ε_{loc} et l'autre global $\varepsilon_{\text{glob}}$, qui correspondent à l'algorithme de point fixe et au nombre de modes à considérer respectivement. Localement, cela revient au même que l'Eq. (1.3). De manière globale, il faut adapter la méthode des moindres carrés mais l'idée reste la même que précédemment.

La limitation de la SVD est qu'elle nécessite de stocker un grand nombre de variables, ce qui peut être rédhibitoire dans le cas de problèmes à grandes dimensions. Par exemple, en considérant 10^2 valeurs par dimension, il y en aura 10^4 à considérer pour deux dimensions, 10^6 pour trois et 10^8 pour quatre.

Une autre limitation de cette méthode est qu'elle ne peut pas être utilisée dans une optique d'interpolation ou d'approximation. En d'autres termes, il est possible de décomposer une solution en plusieurs modes, pour rendre son stockage plus facile, puis de la reconstruire, mais pas de calculer des solutions intermédiaires à celles déjà connues. Par exemple, dans le cas d'une image, elle va être décomposée en vecteurs propres générés par deux fonctions (une par dimension). Dans un second temps, cette image va être reconstruite mais sans apporter d'information supplémentaire : si elle est de mauvaise qualité par manque de pixels, elle le restera. Cela est illustré dans la Fig. 1.10 où deux images sont décomposées puis reconstruites avec respectivement 2, 10 et 100 modes. Dans le cas de la Fig. ??, l'image de reconstruite à l'aide de 100 modes est aussi nette que l'image de

départ, et son coefficient α (Eq. (1.4)) tend vers 0. Dans le cas de la Fig. ??, l'image subit le même traitement à la différence que sa qualité initiale est moins bonne. Le dernier cas prenant en compte 100 modes ne permet pas de retrouver une image nette mais seulement l'image de départ, déjà atteinte avec 10 modes.

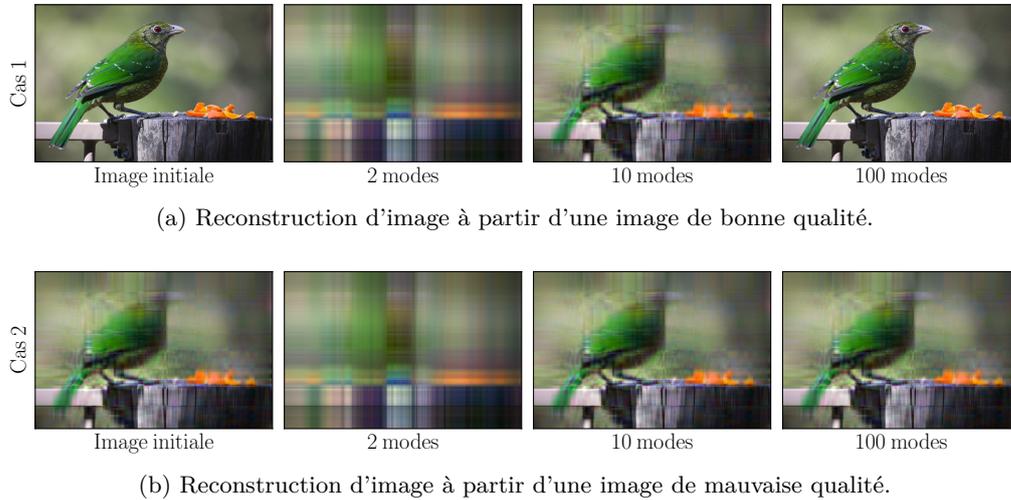


FIGURE 1.10 – Application de la *Singular Value Decomposition* pour un problème de reconstruction d'image

1.4.1.2 Principe de la méthode Proper Orthogonal Decomposition

La méthode POD a été initialement introduite dans l'analyse d'écoulement turbulent par [32] puis plus généralement dans la mécanique des fluides où des cas d'application peuvent être trouvés dans [82, 101, 123]. Son utilisation s'est ensuite diversifiée à d'autres domaines comme la résolution de problèmes dynamiques non-linéaires [80], l'approche système [162] et la reconnaissance de forme [60], la résolution de problèmes inverses [13, 75] ou l'approximation de données [32].

La méthode SVD va décomposer la solution globale en $N \times r$ vecteurs propres, avec N le nombre de dimensions, pour calculer les r premiers modes nécessaires à la reconstruction du modèle. La méthode POD fonctionne dans l'autre sens, tout en se basant sur une SVD initiale. En d'autres termes, les fonctions orthogonales \mathbf{f}_i et \mathbf{g}_i sont déduites d'une base de données initiale à l'aide d'une SVD. Les Nr vecteurs propres sont ensuite construits à l'aide d'un algorithme de point fixe. La solution globale est finalement calculée en prenant en compte un nombre r suffisant de modes pour atteindre le critère de convergence défini en amont, $\varepsilon_{\text{glob}}$.

Comme précédemment, on considère un problème MISO dont la fonction de transfert est donnée par $\mathcal{M} : \mathcal{A} \subseteq \mathbb{R}^2 \longrightarrow \mathcal{B} \subseteq \mathbb{R}$, avec \mathcal{A} le domaine de définition où les variables d'entrée prennent leur valeur et \mathcal{B} , le co-domaine où la sortie évolue. Les solutions de ce modèle sont exprimées par M . La différence entre la SVD et la POD peut alors se résumer

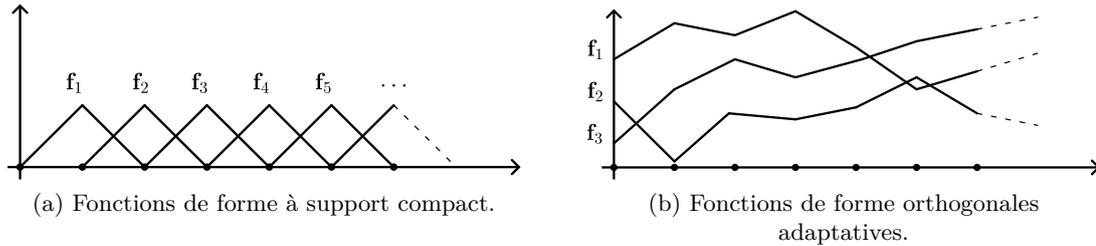


FIGURE 1.11 – Illustration des fonctions de formes utilisées dans la méthode aux Éléments Finis en vis à vis de celles employées dans la méthode *Proper Orthogonal Decomposition*.

par :

$$\begin{cases} \text{M } \underline{SVD} \sum \mathbf{f}_i \otimes \mathbf{g}_i, \\ \text{M } \underline{POD} \sum \mathbf{f}_i \otimes \mathbf{g}_i. \end{cases} \quad (1.12)$$

Pour comprendre le principe de fonctionnement des fonctions orthogonales de la POD, il est intéressant de se référer à la théorie des EF. Dans la méthode classique des EF, des fonctions de forme à support local (approximation 1D, linéaire par morceaux par exemple) vont être utilisées pour approximer et construire la solution globale d'un problème comportant une géométrie maillée : il y aura autant de fonctions de forme que d'éléments discrets dans chaque direction de l'espace. La méthode POD va comporter des fonctions de forme dites orthogonales adaptatives, une par dimension d'entrée. Elles vont évoluer et s'adapter de manière à former les différents modes de la solution par produit tensoriel. Ce parallèle est illustré à la Fig. 1.11, où les deux types de fonctions de forme sont tracées.

Pour illustrer la différence entre la méthodes EF et la POD de manière plus concrète, considérons une personne en mouvement dont seules les positions des articulations sont prises en compte. Avec la méthode EF, l'environnement de la personne correspond à une matrice en 3 dimensions, avec une quantité d'éléments par dimension égale au nombre de points. S'il y a 10 capteurs, la matrice à manipuler aura 1000 éléments. D'un autre côté, la POD permet de considérer seulement les coordonnées dans chaque direction de chaque élément. Ainsi le modèle est décrit par 3 vecteurs composés chacun de 10 éléments, soit 30 éléments en tout. La Fig. 1.12 illustre cette différence.

Ces fonctions orthogonales adaptatives sont construites à partir d'un jeu initial de données, décomposé aux valeurs propres. En conséquence, si le comportement recherché n'est pas inclus dans la base de données de construction (la personne saute au lieu de marcher par exemple), le modèle réduit ne parviendra pas à fournir un résultat correct. En outre, lorsqu'il y a un grand nombre de valeurs propres à considérer, indirectement de modes de la solution, et que le but est seulement d'alléger la solution, la SVD sera plus efficace puisqu'elle ne va pas résoudre le problème, contrairement à la POD dans laquelle des inversions de matrices sont nécessaires (se référer à [92] pour plus d'informations à ce sujet).

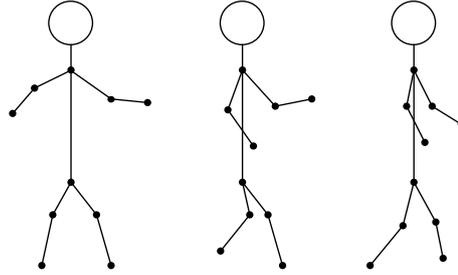


FIGURE 1.12 – Étude du mouvement d’une personne repérée par dix capteurs via la méthode des Éléments Finis ou *Proper Orthogonal Decomposition*. Dans le premier cas, le nombre d’éléments à stocker est de 10^3 contre 3^1 dans le second cas.

Globalement, la méthode POD est une méthode explicite (ou implicite selon la situation) d’approximation déterministe, pouvant aussi être appliquée sur des cas stochastiques. Elle a également un fort caractère intrusif du fait de la décomposition des variables du problème. Notons également qu’il s’agit d’une méthode *a posteriori* nécessitant une première phase d’acquisition de données, avec application d’une SVD.

1.4.1.3 Principe de la méthode Proper Generalised Decomposition

Un des défauts de la méthode POD, qui est commun à la majorité des stratégies de réduction de modèle et de métamodélisations, est la nécessité d’une étape d’apprentissage, dite phase *off-line*. La PGD est une des rares méthodes fonctionnant *a priori* : il n’est pas nécessaire de connaître une solution préalable ou d’avoir une base de données initiale. Les Nr vecteurs propres nécessaires au calcul de la solution sont déterminés par des fonctions de formes évaluées au fur et à mesure de la résolution du problème.

Le premier à avoir introduit une méthode de résolution *a priori* est Ladevèze [102] avec la méthode dite d’*approximation radiale*. Cette dernière est à l’origine du développement du solveur *LArge Time INcrement method* (LATIN) [104, 105]. Initialement employée dans le domaine de la mécanique des milieux continus pour la résolution de problème dynamique à plasticité non linéaire [36, 103], elle s’est étendue à de nombreux domaines comme le calcul d’homogénéisation [106], les études multi-paramétriques [143], la simulation dynamique des chocs [21, 112] ou encore l’optimisation [2, 114]. En outre, ce n’est qu’en 2007 que le terme PGD apparaît dans le travail de Ammar [3] portant sur la résolution d’un problème aux dérivées partielles découlant d’un modèle rhéologique complexe. Pour parvenir à une solution, il applique la méthode d’approximation radiale de [102] qu’il généralise en effectuant une séparation des variables d’entrée du problème.

Dans le nom de la PGD, le terme *Generalised* est employé par opposition au terme *Orthogonal* de la POD, en référence aux types des fonctions de formes employées. Dans la forme classique de la PGD, ces fonctions de formes sont calculées à partir du système d’équations qui décrit le modèle étudié. Cela rend la méthode relativement intrusive. Un cas d’application est présenté dans [20]. Il s’agit d’une étude thermo-mécanique en 3

dimensions d'un matériau composite construit par addition successive de couches. La stratégie décrite dans cet article, faisant référence à la PGD, consiste à séparer le problème thermique du problème mécanique et à décomposer la géométrie 3-D en un système 1-D + 2-D (empilement des couches de composite). Les solutions de ces sous-systèmes sont déterminées à l'aide d'un algorithme de point fixe. La solution finale est ensuite reconstruite par addition de r modes. Pour rendre compte de l'efficacité de la méthode, l'auteur compare les résultats de la méthode PGD à ceux d'une méthode classique basée sur des fonctions de forme linéaires à support compact, i.e., la théorie des EF. La conclusion de cette étude est que la PGD est bien plus rapide que les EF tout en fournissant une erreur faible. Néanmoins, le modèle initial comporte l'hypothèse forte d'avoir des variables séparables.

Ainsi, cette méthode bien que très intéressante et fonctionnant *a priori* possède certaines limites. Il s'agit d'une méthode relativement intrusive travaillant directement sur la formulation du problème. Si ce dernier n'est pas à variables séparables, la PGD est difficilement, voire pas, applicable [35]. En outre, cette méthode peut s'appliquer sur des problèmes d'interpolation, la rendant non-intrusive. Néanmoins, l'avantage de l'absence de la partie *off-line* est perdue puisque les solutions intermédiaires aux points de la base de données doivent être recalculées [65].

1.4.2 Méthodes d'interpolation et d'approximation

La réduction d'ordre permet de traiter des problèmes de grandes dimensions tout en restant peu coûteuse. Néanmoins son intrusivité et la nécessité de décomposer le problème général peuvent la limiter. Selon la situation, le compromis peut être fait d'employer une méthode d'interpolation, ou d'approximation pure pour passer outre cette limite. Il s'agit généralement d'un modèle mathématique reproduisant le comportement d'un système quelconque (physique, mathématique, etc.), alimenté par une base de données de points d'origine quelconque (expérimentations, calculs haute-fidélité, etc.). Les méthodes d'interpolation et d'approximation présentées dans cet état de l'art sont : les régressions déterministes comme la régression linéaire, les régressions intégrant un processus gaussien comme le krigeage, les ANN et la méthode basée sur les NURBS, entités géométriques découlant directement des courbes de Béziérs et *Basis-Spline* (B-spline).

1.4.2.1 Les polynômes du chaos en métamodélisation

Souvent, les méthodes d'interpolation ou d'approximation se basent sur des polynômes ou des sommes ou produits de polynômes. L'objectif ensuite est de calibrer les coefficients dits *coefficients polynomiaux* pour approcher ou interpoler une base de données. Leur performance dépend de leur définition intrinsèque, de la quantité de données d'apprentissage disponible ou encore de la complexité du modèle d'étude, notamment sa non-linéarité, sa

continuité ou sa monotonie. L'expression générale d'un métamodèle construit à partir de polynômes se traduit par :

$$\hat{\mathbf{Y}} = \sum_{i=1}^N \beta_i f(\hat{\mathbf{X}}), \quad (1.13)$$

avec $\hat{\mathbf{Y}}$ le vecteur des outputs du modèle de dimension \mathbb{R}^M et $\hat{\mathbf{X}}$ le vecteur des inputs de dimension \mathbb{R}^N . β_i sont quant à eux les coefficients polynomiaux, généralement déterminés par la minimisation d'une distance entre les points de la base de données d'entraînement \mathbf{Y} et ceux évalués par le métamodèle $\hat{\mathbf{Y}}$. Pour finir, $f(\cdot)$ sont les fonctions polynomiales, ou fonctions de base, dépendantes du modèle mathématiques utilisé. Les plus communes sont les polynômes de Lagrange [46] : les polynômes f_i , définis aux abscisses x_i , avec $i \in [1, \dots, n]$, de degrés $n - 1$ sont exprimés par :

$$f(x)_i = \prod_{i \neq j} \frac{\hat{x} - x_j}{x_i - x_j}. \quad (1.14)$$

Ces polynômes sont assez restreints en tant que métamodèles, notamment parce qu'ils traitent mal les non-linéarités et que leur degré dépend du nombre de points à interpoler. D'autres polynômes existent comme les polynômes de Newton utilisés dans l'interpolation de Hermite [46], les polynômes de Bernstein utilisés dans la construction des courbes de Bézier et leurs dérivées [46, 138] ou encore les polynômes du chaos [107]. Ces derniers font intervenir dans la formulation même des fonctions de bases un terme stochastique. Il permet de récupérer des informations sur la variance et le comportement statistique du modèle en fonction de ses paramètres et de leur plage de variation. Ainsi, ces polynômes sont employés dans deux cadres des mathématiques, souvent liés : l'analyse de sensibilité paramétrique et la métamodélisation. L'expression d'un métamodèle liés aux polynômes du chaos se fait par :

$$\hat{\mathbf{Y}} = \sum_{i=1}^N \beta_i \phi(\hat{\mathbf{X}}), \quad (1.15)$$

où ϕ sont les fonctions de base exprimées suivant la même démarche qu'une décomposition de la variance. Ainsi, en utilisant les polynômes du chaos en métamodélisation, il est également possible de déduire de leur expression les indices de Sobol présentés plus haut. L'emploi des polynômes du chaos en métamodélisation est en cours de développement avec notamment l'apparition des méthodes de *re-sample polynomial chaos expansion* [188], de *high dimensional model reduction - polynomial chaos expansion* [191], de *polynomial chaos expansion net* [160] ou encore, de *co-kriging based on polynomial chaos expansion* [48].

Néanmoins, les PECs dans la métamodélisation comportent pour le moment un certain nombre de limitations : les inputs doivent être indépendants, si le problème est complexe

il faudra une base de données très riche pour avoir un bon comportement du terme stochastique [188].

1.4.2.2 Régression déterministe - Régression linéaire

Méthodes issues de la manipulation de PE, les régressions correspondent à un ensemble de méthodes statistiques utilisées pour approcher le comportement d'une variable de sortie en fonction de variables d'entrée, complétées par un terme d'erreur par rapport à la solution exacte, de la même manière que les polynômes [12]. Si cette relation est linéaire, la régression sera du même type. En considérant un problème MISO dont la fonction de transfert est donnée par $\mathcal{M} : \mathcal{A} \subseteq \mathbb{R}^2 \rightarrow \mathcal{B} \subseteq \mathbb{R}$, avec \mathcal{A} le domaine de définition où les variables d'entrée prennent leur valeur et \mathcal{B} , le co-domaine où la sortie évolue, un processus de régression s'exprime par :

$$\hat{y} = \mu(\hat{x}_1, \hat{x}_2) + \varepsilon, \quad (1.16)$$

où \hat{y} correspond à l'espérance μ de la nouvelle sortie estimée sur un site non échantillonné (\hat{x}_1, \hat{x}_2) et ε correspond à l'écart entre les données y et le métamodèle \hat{y} . Le premier terme fait référence à la partie déterministe du métamodèle régressif là où ε peut être déterministe ou stochastique selon sa méthode d'évaluation. Dans le cas de la régression linéaire classique, ce terme sera déterministe, noté ϵ , et le but va être à la fois de le minimiser et d'exprimer μ sous la forme d'un polynôme de la forme :

$$\mu(x_1, x_2) = \sum_{i+j \leq p} \beta_{i,j} \hat{x}_1^i \hat{x}_2^j, \quad (1.17)$$

où p est le degré maximal que peuvent prendre les variables du modèle et $\beta_{i,j}$ les coefficients polynomiaux associés respectivement aux termes du polynôme. Tout l'enjeu d'une régression linéaire va être de fixer le degré du polynôme et de calibrer les valeurs des coefficients $\beta_{i,j}$. Ces étapes se font dans la partie *off-line* du métamodèle, à partir d'une base de données composées de solution y estimées sur des sites échantillonnés (x_1, x_2) . Le degré p ne dépasse généralement pas 3, qui est un compromis entre précision, complexité et bruit polynomial du métamodèle. Quant aux coefficients $\beta_{i,j}$, ils sont estimés via la minimisation du terme d'erreur ϵ . Pour y parvenir, plusieurs méthodes existent comme celles du maximum de vraisemblance [171] ou encore de la minimisation au sens des moindres carrés. Cette dernière consiste à minimiser la distance euclidienne (ou norme L^2) entre les solutions de la base de données y et les solution recalculées \hat{y} :

$$\beta_0 = \underset{\beta \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} \|\epsilon_n\|_2 = \underset{\beta \in \mathbb{R}^{p \times p}}{\operatorname{argmin}} \|y_n - \hat{y}_n\|_2, \quad (1.18)$$

avec n le nombre de points de la base de données. Pour plus d'informations à ce sujet ainsi que pour des cas d'applications, le lecteur est invité à se référer à [110]. Ce type de

métamodèle est intéressant pour sa simplicité de fonctionnement mais reste efficace sur des problèmes linéaires ou peu non-linéaires. Pour pouvoir traiter des problèmes plus complexes, des méthodes utilisant une formulation stochastique du terme d'erreur ε existent, comme le krigeage, définissant ce terme comme un processus gaussien.

1.4.2.3 Régression stochastique - Processus gaussien et krigeage

La stratégie dite de *krigeage* est un outil mathématique d'interpolation qui a été théorisée par le français Matheron [118, 119] en 1962-1963. Ce dernier a nommé ce procédé *krigeage* en référence à Krigde, première personne ayant travaillé sur cette méthode sans l'avoir complètement théorisée. Par définition, il s'agit d'une méthode stochastique d'estimation linéaire garantissant un minimum de variance. Cette méthode va interpoler les points de la base de données y_n qui lui sont fournis et donner une estimation de la variance entre deux sites d'interpolation. D'un point de vu plus formel, en reprenant l'Eq. (1.16), un processus de krigeage s'exprime par [12] :

$$\hat{y} = \mu(\hat{x}_1, \hat{x}_2) + \varepsilon, \quad (1.19)$$

où $\mu(\cdot)$ correspond à l'espérance de la base de données et ε à une fonction stochastique gaussienne : elle est aléatoire stationnaire, d'espérance nulle et de structure de dépendance connue. Cette dernière propriété permet de prendre en compte à la fois la distance entre les données y et celles du métamodèle \hat{y} , mais également les distances entre les points y . Ainsi, pour définir complètement un processus de krigeage, il est nécessaire de spécifier l'expression de la tendance μ mais également la forme de la dépendance du métamodèle à ε . Nous présentons ci-dessous certaines expressions de μ , définissant le type du krigeage :

- Si $\mu(x_1, x_2) = m$ une constante connue, alors il s'agit d'un krigeage *simple* ;
- Si $\mu(x_1, x_2) = \mu$ une constante inconnue, alors il s'agit d'un krigeage *ordinaire* ;
- Si $\mu(x_1, x_2) = \sum_{i+j \leq p} \beta_{i,j} f_{i,j}(x_1, x_2)$ une combinaison linéaire de fonctions, alors il s'agit d'un krigeage *universel*.

La seconde étape de création d'un métamodèle évoluant selon un processus de krigeage consiste à déterminer le comportement de ε . Pour ce faire, il est possible de se référer au variogramme de la base de donnée, regroupant les variances locales entre deux valeurs y_n et y_{n+1} . On peut relever ici qu'une régression linéaire correspond à un krigeage universel dont ε est considéré indépendant de la structure spatiale de la base de données et pour laquelle μ est exprimée suivant un polynôme.

Pour illustrer le fonctionnement d'un processus de krigeage, la Fig. 1.13 trace la métamodélisation de la fonction $y = x \sin(x)$. Les données d'entraînement sont représentées en rouge, celles issues du métamodèles \hat{y} en noir et les variances en nuances de vert (nulles aux points interpolés). Cette figure illustre également que l'enrichissement de la base d'entraî-

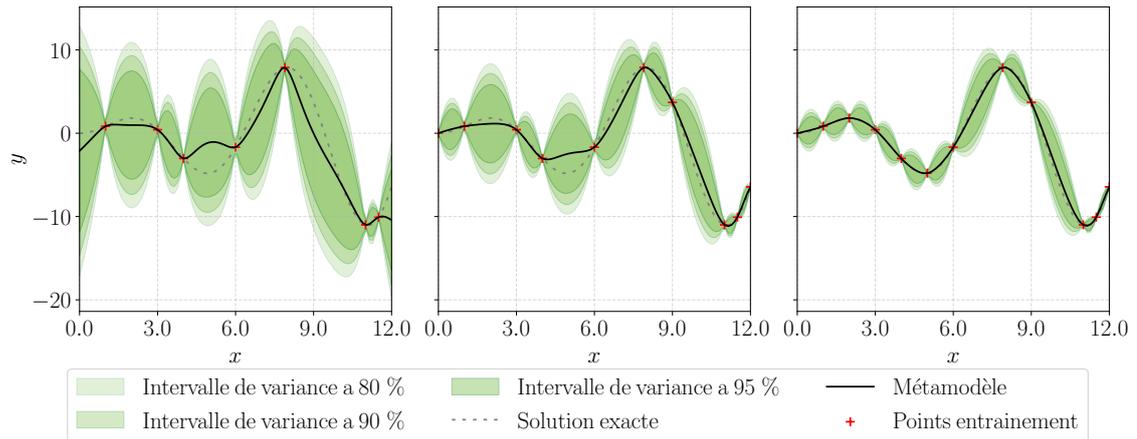


FIGURE 1.13 – Illustration d’une interpolation par krigeage universel sur des points y_n calculés par l’application $x_n \rightarrow x_n \sin(x_n) : \mathcal{A} \subseteq \mathbb{R} \rightarrow \mathcal{B} \subseteq \mathbb{R}$, avec \mathcal{A} le domaine de définition où le paramètre d’entrée prend ses valeurs et \mathcal{B} , le co-domaine où la sortie évolue [14]. De gauche à droite, la base de données est enrichie au niveau des zones de plus forte variance pour diminuer celle-ci.

nement, où la variance est la plus importante, permet de réduire l’erreur entre la fonction initiale et le métamodèle, ainsi que la répartition de la variance.

Le krigeage se présente comme étant le meilleur estimateur linéaire non-biaisé. Pour pouvoir être applicable à des problèmes de différents types, d’autres méthodes basées sur cette formulation existent comme celle prenant en compte les erreurs de mesure [41], celle avec des dérivées externes [69] ou encore celle à plusieurs variables, appelé co-krigeage [182]. Néanmoins, cette méthode comporte certaines limites :

- Propriété de support global. Si un point de la base de données est modifié, tout le métamodèle le sera également ;
- Elle est applicable sur des problèmes à faible dimension, i.e., inférieur à 10 ;
- Si la base de données est trop complexe (fortes non-linéarités par exemple) le terme stochastique ε sera difficile, voire impossible, à exprimer.

Pour plus d’information sur le sujet le lecteur est invité à se référer à [12] et à [182] pour des exemples d’application.

1.4.2.4 Réseaux de neurones

Cette méthode s’appuie sur le proverbe *la qualité se trouve dans la quantité*. Il s’agit également de son principal défaut : pour pouvoir avoir de bon résultats, la phase *off-line* de l’algorithme est souvent très coûteuse en ressources et temps de calcul. Les premiers travaux à ce sujet sont apparus dans les années 1950 [113]. Ils sont construits selon un paradigme biologique, celui du neurone formel, correspondant au fonctionnement mathématique d’un neurone naturel : une fonction qui renvoie une sortie calculée selon les paramètres d’entrée fournis (notion introduite par Lettvint [113]). Partant de ces travaux,

une multitude de variantes d'ANN ont vu le jour, se différenciant selon plusieurs critères et en grande partie répertoriés dans [71, 77, 121].

Chaque ANN a son propre mode de fonctionnement. Néanmoins, ils sont construits tous de la même manière : plusieurs couches communiquent entre elles par le biais de fonctions définies en amont, i.e., les neurones artificiels. La première couche, la couche des entrées, comporte les variables qui vont évoluer sur leur plage de variation définie en amont. La dernière couche est composée des sorties du problème : elle rassemble tous les résultats des fonctions précédentes. Les couches intermédiaires, quant à elles, sont composées de fonctions qui vont permettre de reproduire le comportement du modèle initial. Ces couches vont travailler de proche en proche, en fournissant les variables de sortie en entrée de la couche suivante, toujours dans le sens de résolution : il n'y pas de rétroaction parmi les couches comme cela peut être le cas entre les sous-systèmes d'une approche MBSE par exemple. Pour pouvoir construire et utiliser un réseau de neurones, plusieurs étapes sont nécessaires :

1. La définition des fonctions intrinsèques au problème qui jouent le rôle de neurone ;
2. La sélection de l'architecture même du réseau. Cela consiste à déterminer principalement le nombre de couches et de fonctions par couche à prendre en compte ;
3. L'apprentissage durant la partie *off-line*, qui va permettre de déterminer les chemins entre les neurones qui fournissent la meilleure solution. Cette étape est la plus coûteuse notamment parce que l'algorithme doit trouver tous les chemins, les optimaux mais également ceux qui ne le sont pas.

Ce fonctionnement est illustré dans la Fig. 1.14. Il s'agit d'un modèle ANN à deux couches intermédiaires comportant respectivement 5 et 4 neurones. A l'état initial, tous les chemins sont considérés comme solutions possibles. A l'état final de la partie *off-line*, les chemins menant à une solution sont retenus tandis que les autres non. Pour cela, $4 \times 5 \times 4 \times 3 = 240$ chemins ont dû être testés, avec autant de données que nécessaire.

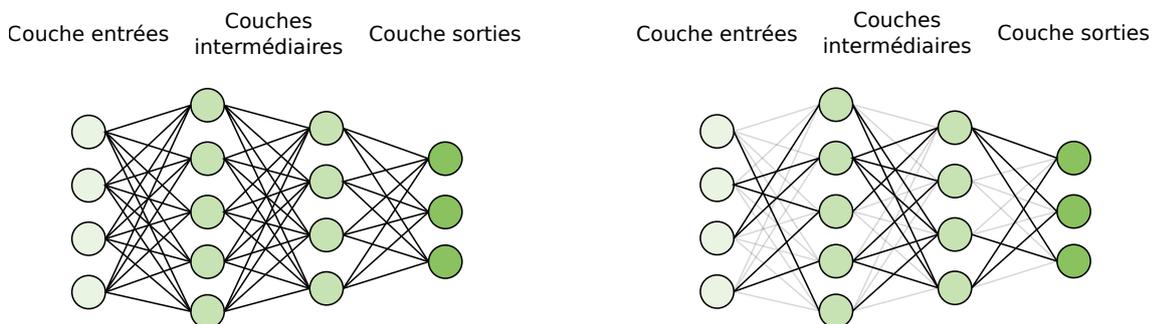


FIGURE 1.14 – Représentation d'un réseau de neurones à 2 sous-couches, 4 entrées et 3 sorties. La figure de gauche correspond au réseau dans son état initial, et la figure de droite au réseau une fois les chemins ne menant pas à une solution écartés.

L'avantage principal des ANN est la possibilité de prendre en compte un très grand

nombre de variables et de fonctions, les rendant très polyvalents sur les types de cas d'application. De plus, une fois construit, le problème va être résolu très rapidement et avec une grande précision. Néanmoins, ce type de métamodèle propose des solutions qui ne sont pas nécessairement répétables et ne peuvent pas être optimisés simplement par des algorithmes d'optimisation sous contraintes par descente de gradient : il n'est pas évident de déterminer leur gradient étant donné qu'ils ne sont pas décrits par une équation analytique. Pour plus d'informations à ce sujet, le lecteur peut se référer à [71, 77, 192, 193].

1.4.2.5 Généralité sur les hypersurfaces *Non-Uniform Rational Basis-Spline*

Les NURBS sont des entités géométriques, généralisant les B-splines, elles-mêmes généralisant les courbes de Bézier [138]. Elles tirent leurs origines dans la CAO de l'industrie automobile dans les années 1990. Initialement employées pour représenter des formes géométriques réelles, i.e., de 1 à 3 dimensions, elles ont trouvé de nouvelles applications dans de nouveaux domaines comme l'optimisation topologique [29, 38, 39, 133, 149], l'optimisation de forme [17, 128], l'optimisation du champ d'anisotropie pour les composites à rigidité variable [126, 127], et la génération de métamodèles [9–11], objet de ce travail.

Considérons un problème MIMO dont la fonction de transfert est donnée par $\mathcal{M} : \mathcal{A} \subseteq \mathbb{R}^N \longrightarrow \mathcal{B} \subseteq \mathbb{R}^M$, avec \mathcal{A} le domaine de définition de dimension N où les paramètres d'entrée prennent leur valeur et \mathcal{B} , le co-domaine de dimension M où les sorties évoluent. L'objectif d'un métamodèle NURBS est d'approcher une base de données \mathbf{Q} de $N_{\text{TP}} = \prod_{k=1}^N (r_k + 1)$ *Target Points* (TPs), avec $(r_k + 1)$ le nombre de TPs dans la direction k , de dimension M (i.e., chaque *Target Point* (TP) est défini dans \mathbb{R}^M), venant de résultats d'expérimentation ou de calculs haute-fidélité. Formellement une entité NURBS est exprimée par :

$$\mathbf{h}(\zeta_1, \dots, \zeta_N) = \frac{\sum_{i_1=0}^{n_1} \dots \sum_{i_N=0}^{n_N} N_{i_1, s_1}(\zeta_1) \dots N_{i_N, s_N}(\zeta_N) \boldsymbol{\omega}_{i_1, \dots, i_N} \mathbf{P}_{i_1, \dots, i_N}}{\sum_{j_1=0}^{n_1} \dots \sum_{j_N=0}^{n_N} N_{j_1, s_1}(\zeta_1) \dots N_{j_N, s_N}(\zeta_N) \boldsymbol{\omega}_{j_1, \dots, j_N}}, \quad (1.20)$$

avec \mathbf{h} l'hypersurface NURBS possédant M coordonnées, évaluées aux coordonnées normalisées $(\zeta_1, \dots, \zeta_N)$ et $\mathbf{P}_{i_1, \dots, i_N}$ le vecteur contenant les M coordonnées du *Control Point* (CP) générique. L'hyper-réseau de contrôle est composé de $N_{\text{CP}} = \prod_{k=1}^N (n_k + 1)$ CPs, avec $(n_k + 1)$ le nombre de CPs dans la direction k . Pour chaque CP, un poids $\omega_{i_1, \dots, i_N} > 0$ est associé. Pour finir $N_{i_k, s_k}(u^{(k)})$ sont les polynômes de Bernstein de degré s_k , associés à la direction d'entrée k . Ces polynômes sont définis récursivement par l'algorithme de Cox et De Boor [40] en fonction des polynômes de degré inférieur $0 \leq q_k < s_k$. Ces polynômes sont définis non nuls sur $s_k + 1$ intervalles du vecteur nodal. Ce dernier est déduit à partir des coordonnées normalisées ayant généré la base de données de départ. Les théories et algorithmes permettant de calculer les points \mathbf{h} d'une entité NURBS seront détaillés plus loin, dans le chapitre 3. Cet état de l'art propose seulement une caractérisation qualitative

de ces entités géométriques.

Les NURBS sont employées dans ce travail pour approximer le comportement de modèles MIMO. Dans certains cas particuliers, elles correspondent à des objets mathématiques pouvant être représentés graphiquement (usage initial en CAO). C'est le cas des courbes planes, $N = 1$ et $M = 2$, des courbes volumiques, $N = 1$ et $M = 3$, des surfaces, $N = 2$ et $M = 3$ ou encore des volumes, $N = 3$ et $M = 3$. Dans les autres cas nous parlerons, parfois par abus de langage, d'*hypersurfaces*. Ces entités sont intéressantes pour leur capacité à s'adapter facilement à des problèmes de complexité différente, sans hypothèses préalables. Pour cela, le métamodèle en question est piloté par quatre paramètres globaux :

- le degré s_k des fonctions de base, fixé en général à 2 ou à 3 pour un bon compromis entre précision, stabilité et quantité de données requises à l'approximation au plus juste ;
- les CPs, formant un réseau de contrôle, permettent de diriger la NURBS. Cette dernière étant une entité convexe, elle sera toujours à l'intérieur du réseau. Les Fig. 1.15a et Fig. 1.15b illustrent cela sur un problème de type courbe plane, en plaçant respectivement $\mathbf{p}_6 = (7.854, 12.017)$ et $\mathbf{p}_6 = (7.854, 5)$;
- les poids ω_{i_1, \dots, i_N} associés aux CPs. Selon leur valeur unitaire par rapport à leur moyenne globale, ils peuvent attirer ou repousser l'hypersurface NURBS. Ce phénomène est illustré par les Fig. 1.16a et Fig. 1.16b où $\omega_6 = 10$ et $\omega_6 = 0.1$ respectivement, comparé au cas où $\omega_6 = 1$;
- les intervalles de définition des fonctions de base $N_{i_k, s_k}(u^{(k)})$. Chacune de ces fonctions évolue sur un intervalle défini par un vecteur nodal. L'organisation de ce vecteur va permettre de donner plus ou moins d'importance à une zone de l'espace des outputs en rapprochant deux nœuds successifs. Cela est illustré par les Fig 1.17a et Fig 1.17b où les nœuds v_9 et v_{10} ont été placés aux coordonnées 0.75 et 0.875 puis aux coordonnées 0.90 et 0.95. En outre, un vecteur nodal associé à une NURBS comporte $s_k + 1$ nœuds égaux à 0 et de même, $s_k + 1$ nœuds égaux à 1. Cela permet à l'entité géométrique d'interpoler le premier et le dernier TP de la base de données.

Dans certains cas d'utilisation, une hypersurface NURBS peut correspondre à des entités mathématiques plus simples : il s'agit d'une généralisation de celles-ci. Par exemple, si tous ses poids sont fixés à 1, alors l'entité correspondante est une B-spline et son expression peut se simplifier par :

$$\mathbf{h}(\zeta_1, \dots, \zeta_N) = \sum_{i_1=0}^{n_1} \dots \sum_{i_2=0}^{n_2} N_{i_1, s_1}(\zeta_1) \dots N_{i_N, s_N}(\zeta_N) \mathbf{P}_{i_1, \dots, i_N}. \quad (1.21)$$

Par ailleurs, si on ne considère qu'une seule fonction $N_{i_k, s_k}(u^{(k)})$ définie sur le vecteur $\mathbf{v}^{(k)} = [1, 1, 1, 1, 0, 0, 0, 0]$, avec $s_k = 3$, cela revient à utiliser une courbe de Bézier [46].

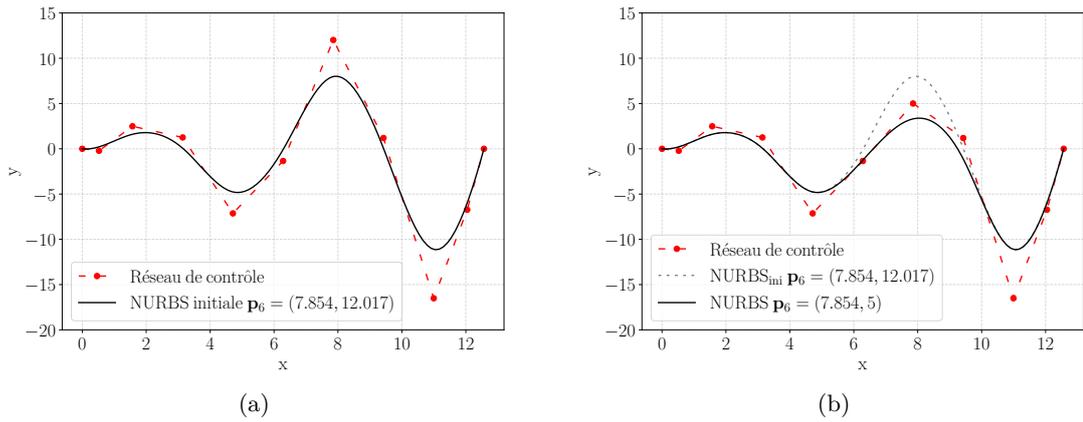


FIGURE 1.15 – Influence des points de contrôle sur une courbe NURBS. En (a) $\mathbf{p}_6 = (7.854, 12.017)$ et en (b) $\mathbf{p}_6 = (7.854, 5)$.

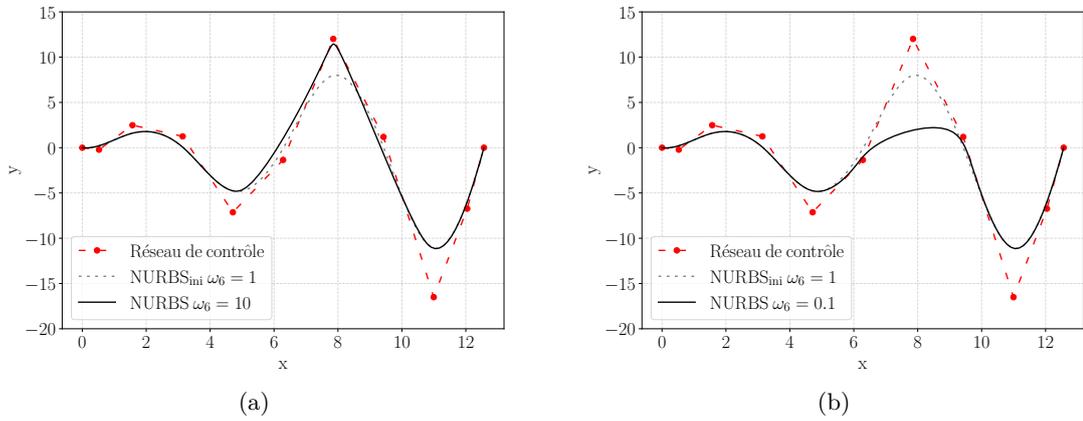


FIGURE 1.16 – Influence des poids sur une courbe NURBS. En (a) le poids $\omega_6 = 10$ et en (b) $\omega_6 = 0.1$, respectivement comparé au cas où tous les poids sont fixés à l'unité.

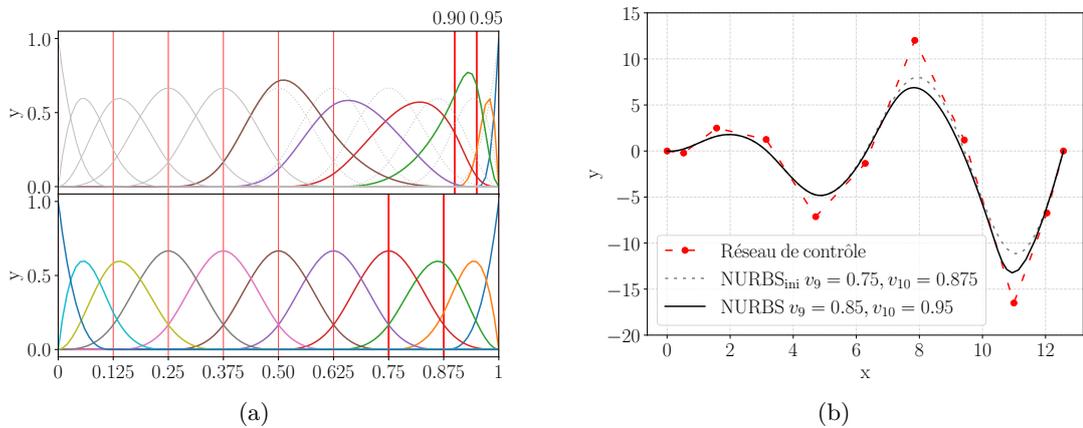


FIGURE 1.17 – Influence du vecteur nodal sur une courbe NURBS. En (a) la courbe tracée avec un vecteur nodal uniforme et en (b), les nœuds v_9 et v_{10} ont été placés aux coordonnées 0.75 et 0.875 puis aux coordonnées 0.90 et 0.95.

En résumé, les NURBS sont des entités géométriques qui ont été étendues à la méta-modélisation [9–11]. Elles sont complètement indépendantes de la formulation du problème sur lequel elles s’appliquent et possèdent une propriété de support local très intéressante. Par exemple en présence d’une base de données ayant un grand intervalle de variation mais une faible variance, l’hypersurface ne sera pas trop impactée par les données extrêmes. Néanmoins, si le nombre de dimensions du problème étudié devient trop grand, la partie *off-line* du métamodèle va devenir très importante. De plus, en cas de non-linéarités, si la base de données génératrice ne comporte pas de points au niveau de ces non-linéarités, l’entité NURBS ne pourra pas les prédire.

1.4.3 Caractéristiques

Les méthodes de métamodélisation présentées dans cet état de l’art ont toutes leurs avantages et leurs inconvénients selon le type de problème traité. La Fig. 1.18 propose de les classer selon le poids de leurs parties *off-line* et *on-line*, leur polyvalence, i.e., si elles sont adaptables à des problèmes non-linéaires, et leur caractère intrusif. Les méthodes basées sur les ANN sont les plus performantes en termes d’utilisation et de traitement de problèmes non-linéaires. Elles permettent également de traiter des problèmes à très grandes dimensions d’entrée et de sortie. Néanmoins, le prix à payer est la nécessité d’avoir une base de données très importante pour l’entraînement du réseau. Pour parvenir à une telle polyvalence tout en réduisant le coût de la partie *off-line*, sans pour autant parvenir à la supprimer comme pour la PGD, cette thèse propose l’emploi d’une méthode basée sur les entités NURBS [9–11]. Cependant, pour être performante, cette méthode nécessite de fixer *a priori* un certain nombre de paramètres. Des lois empiriques existent, présentées par [178], mais ne sont pas systématiques. Plus récemment, [10] et [17] ont proposé de créer un métamodèle NURBS en générant les paramètres globaux de cette entité via un algorithme génétique et une descente de gradient. Néanmoins, dans le premier cas, la méthode se restreint à des problèmes de *surface-fitting* à dimensions réduites. Dans le second cas, les gradients sont calculés par des méthodes classiques, pas ou peu adaptées au cas traité. Ces travaux proposent une généralisation de la méthode de [17] à des cas MIMO, présentée en détail au chapitre 3.

1.5 Conclusion et stratégie complète

L’objectif de ce travail de thèse est de développer une stratégie de modélisation multifidélité via une approche MBSE incluant un métamodèle basé sur les hypersurfaces NURBS : la méthode MHMN. Le but de cette stratégie est de compléter et d’approfondir les études de pré-dimensionnement réalisées en amont de projet, par un bureau d’études ou des architectes produit. Les méthodes MBSE classiques sont très efficaces dans le pré-dimensionnement d’un système multiphysique, mais peuvent présenter des la-

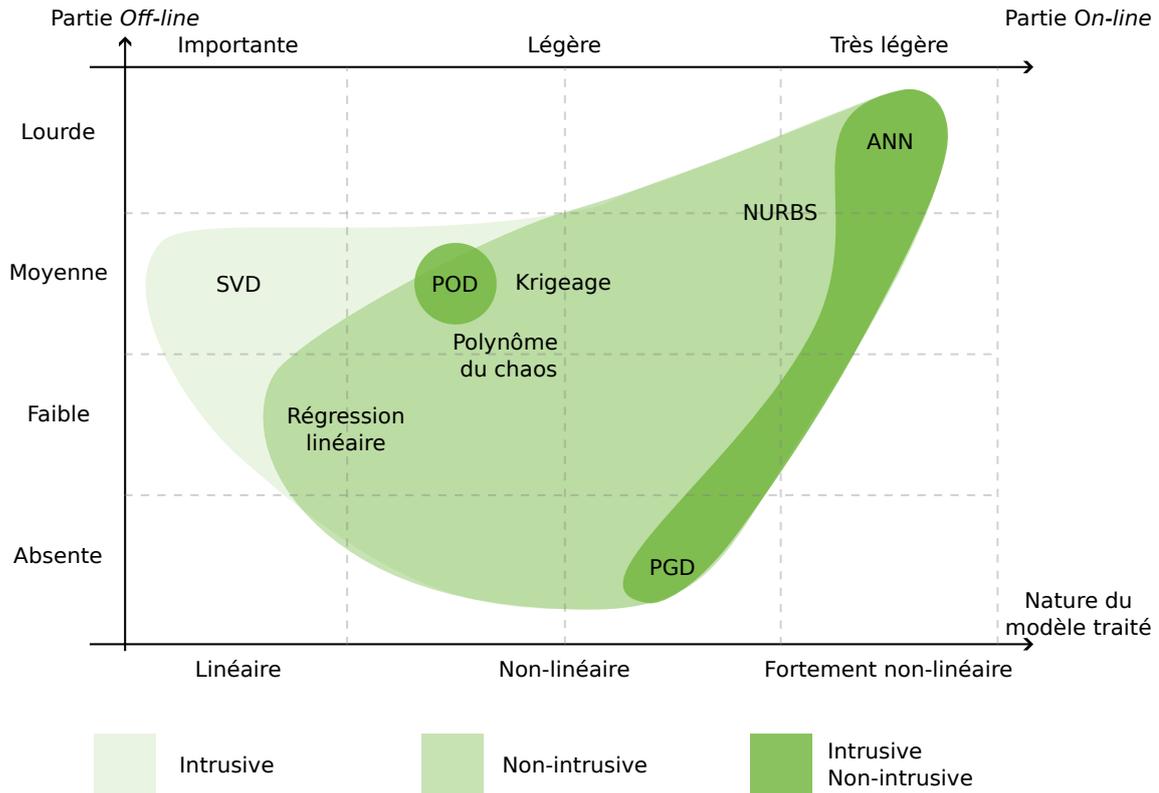


FIGURE 1.18 – Caractéristiques des différentes méthodes de métamodélisation évoquées.

cunes en précision et en discrétisation de l'espace. L'idée est de compléter cette approche en remplaçant les sous-systèmes critiques et/ou influents par un métamodèle haute fidélité basé sur une hypersurface NURBS. L'objectif final est d'employer avec parcimonie les calculs haute-fidélités, ou les expérimentations, tout en ayant une modélisation globale peu coûteuse du dispositif d'étude. Cette méthodologie peut être comparée à des approches globales / locales comme celles présentées dans [27, 109] mais dans un contexte de pré-dimensionnement.

Le choix du langage MBSE s'est orienté vers Modelica pour son aspect *open source* mais également pour sa facilité de communication avec d'autres logiciels et son fonctionnement acausal. Sa capacité à être exporté selon la norme FMI permet de réaliser une grande quantité de calculs en un temps réduit, très utile pour réaliser des analyses statistiques comme l'ASG. Cet outil mathématique, couplé à un avis critique d'ingénieur, permet dans un second temps d'identifier les sous-systèmes influents ou critiques du système global. Un sous-système influent est un modèle qui influe grandement sur la sortie. Un sous-système critique est un modèle identifié par un retour d'expérience ou un avis critique d'ingénieur, qui influe sur le comportement de la sortie, sans être systématiquement identifié par une ASG. Le chapitre 2, qui a été extrait de la publication [180], propose, à titre d'illustration, l'application de méthodes d'ASG sur des modèles Modelica exportés au format FMU.

Une fois le, ou les, sous-système d'intérêt identifié, il est remplacé par un métamodèle.

Ce métamodèle doit comporter plusieurs caractéristiques pour rentrer dans le cahier des charges de la stratégie de modélisation multifidélité :

- son utilisation, i.e., sa partie *on-line*, doit être rapide et peu coûteuse pour pouvoir être appelée en temps réel par le modèle global, sans qu’il n’y ait de stockage des résultats par le métamodèle ;
- il doit pouvoir être applicable à des problèmes MIMO non-linéaires, non-monotones voire discontinus ;
- sa phase d’entraînement, i.e., sa partie *off-line*, doit rester raisonnable. En d’autres termes, il ne faut pas que la base de données génératrice du modèle réduit soit aussi importante que la quantité de données demandées à celui-ci ;
- une fois construit, il doit pouvoir être utilisé avec peu de ressources. Son poids numérique doit être faible et s’adapter à l’environnement numérique d’emploi : il doit pouvoir être employé de la manière la plus autonome possible.

Il n’existe pas de méthodes uniques respectant parfaitement tous ces critères. Le compromis a donc été d’employer la stratégie de métamodélisation basée sur les entités NURBS. Ces entités géométriques, bien que largement répandues dans le domaine de la conception, sont encore très peu utilisées dans d’autres domaines. Néanmoins, depuis plusieurs années, le laboratoire I2M s’intéresse aux propriétés des NURBS, et développe des stratégies de calculs et de modélisation basées dessus. Peuvent être citées des stratégies d’optimisation topologique [38, 39, 149], d’optimisation de forme [17, 128], d’optimisation du champ d’anisotropie pour les composites à rigidité variable [126, 127] ou encore de métamodélisation [9–11]. Fort de cette expérience, cette thèse propose la généralisation de la méthodologie de métamodélisation basée sur les entités NURBS comme la solution d’un problème d’optimisation sous contraintes, applicables à des problèmes MIMO. Nous montrerons également que cette méthode est un bon compromis parmi les méthodes de métamodélisation évoquées dans la section précédente.

L’étape finale de la stratégie développée dans ce travail, est d’intégrer le métamodèle dans le modèle MBSE global. Ainsi, un parallèle entre les langages Modelica (modèle MBSE) et Python (métamodèle NURBS) est fait. L’intégration du code Python de métamodélisation dans le modèle global MBSE est permise par la méthodologie développée par Dassault System [167]. Elle consiste à faire communiquer les deux modèles par l’échange de variables discrètes et continues à chaque pas de temps de la résolution via un logiciel de gestion de bibliothèques C. Ainsi, à chaque pas de temps, le modèle Modelica fournit au modèle réduit un vecteur d’inputs. Le modèle réduit fournit en retour un vecteur d’outputs exploitable par Modelica. La mise en place de la stratégie complète est décrite par le schéma dans la Fig. 1.19.

Le chapitre 3, décrit formellement la construction d’un métamodèle basé sur les entités NURBS et la formulation du problème d’optimisation associée et la stratégie de résolution



FIGURE 1.19 – Illustration des étapes principale de la mise en place de la stratégie Modèle Hybride *Model-Based System Engineering* NURBS.

de ce dernier. Le chapitre 4 quant à lui illustre l'utilisation de ce métamodèle sur des *benchmarks* représentatifs, plus ou moins complexes. Pour finir, le chapitre 5 propose une mise en place de la stratégie complète développée dans ce travail de thèse, avec une partie des résultats et réflexions préliminaires obtenus et rencontrés tout au long de cette période de doctorat.

2

Identification des sous-systèmes critiques et influents

Sommaire

2.1	Introduction au chapitre	53
2.2	Analyse de sensibilité	53
2.2.1	Indices de Sobol et analyse de la variance	53
2.2.1.1	Algorithme de Saltelli	57
2.2.1.2	Méthode <i>Fourier Amplitude Sensitivity Test</i>	57
2.2.2	Indices de Shapley	58
2.3	Problème analytique de référence : la fonction d'Ishigami	58
2.4	Problèmes non-linéaires dynamiques - Approche MBSE	62
2.4.1	L'amortisseur de chocs	62
2.4.1.1	Description analytique	62
2.4.1.2	Modèle MBSE	64
2.4.1.3	Résultats numériques	66
2.4.2	Oscillateur à frottement sec	68
2.4.2.1	Description analytique	68
2.4.2.2	Modèle MBSE	69
2.4.2.3	Résultats numériques	69
2.5	Analyse de sensibilité globale sur des modèles EF	72
2.5.1	Problème 1 : Analyse statique d'une structure treillis	72
2.5.1.1	Formulation du problème et modèle numérique	72
2.5.1.2	Résultats numériques	72
2.5.2	Problème 2 : Analyse de flambement par valeurs propres d'un panneau raidi en composite mono-raisseur.	74
2.5.2.1	Formulation du problème et modèle numérique	74
2.5.2.2	Résultats numériques	77
2.6	Conclusion du chapitre	79

RÉSUMÉ DU CHAPITRE

Ce chapitre porte sur l'application et la comparaison de méthodes d'analyse de sensibilité appliquées à des modèles d'intérêt. Il a fait l'objet d'un article publié début 2023 [180]. Le modèle d'un système est considéré comme étant une fonction de transfert qui exprime des sorties en fonction de variables d'entrée pour reproduire le comportement du système. Une analyse de sensibilité est un outil mathématique employé pour classer ces variables d'entrée en fonction de leur influence sur la (ou les) sortie d'intérêt.

Pour réduire son coût d'étude et son temps de calcul, une solution est d'identifier les variables les plus influentes sur la, ou les, sortie d'intérêt. Il est alors possible de se consacrer à la calibration précise des variables identifiées via une optimisation ou une étude plus approfondie de leur comportement au sein du système global. Ce chapitre propose la comparaison de différents algorithmes de calculs de l'influence des paramètres de conception avant de les appliquer à des problèmes d'intérêt. Pour revenir aux applications visées dans le cadre de ce travail de thèse, ce chapitre propose l'application des Analyses de Sensibilité Globale basées sur les indices de Sobol et l'effet Shapley sur des systèmes modélisés par approche Model-Based System Engineering (MBSE) et Éléments Finis (EF).

2.1 Introduction au chapitre

Une AS permet de déterminer les variables d'entrée qui influent le plus sur les sorties d'intérêt du système d'étude. Par définition, l'influence d'un paramètre définit dans quelle mesure ce paramètre impacte le résultat du problème lorsque sa valeur est modifiée. Il s'agit, dans la majorité des applications, d'étudier le comportement de la variance de la sortie. Selon la méthode considérée, les résultats de l'AS sont qualitatifs ou quantitatifs (se référer au chapitre 1). Dans le cadre de ce travail, seules des méthodes quantitatives ont été considérées. Une première phase d'étude consiste à étudier le taux de convergence, i.e., le nombre de simulations nécessaire pour atteindre des résultats fiables, des principaux algorithmes d'ASG quantitatives disponibles. Les algorithmes en questions permettent de calculer les indices de Sobol (algorithmes de Saltelli de 2002 [151] et de 2010 [152]) ainsi que les indices de Shapley (algorithme proposé par Goda [66], adapté à l'environnement Python (annexe C.1)). Les conclusions de cette étude de convergence ont ensuite permis d'appliquer efficacement ce type de méthodes sur des problèmes d'ingénieur d'intérêt. Les deux premiers cas tests, modélisés par approche MBSE, sont étudiés pour leurs fortes non-linéarités dues à des comportements instables qui peuvent induire des discontinuités dans le temps. Il s'agit de systèmes oscillants, l'un soumis à une perte de charge singulière non-linéaire, l'autre à une force de frottement à seuil. Un travail réalisé en parallèle de ces ASG de modèle MBSE, a été de montrer l'efficacité de la norme FMI pour mener à bien un grand nombre de simulations en un temps raisonnable. En outre, deux autres cas tests sont étudiés et modélisés par la méthode des EF. Il s'agit d'un modèle de poutre en treillis dont la variation de section des poutres influence l'équilibre de la structure, et d'un modèle de panneau raidi mono-raideur en flambement.

A cette fin, le chapitre est organisé comme suit. La section 2.2 rappelle la formulation théorique des indices de Sobol et de l'effet Shapley. La section 2.3 présente la comparaison entre les différents algorithmes de calcul des indices de Sobol et de l'effet Shapley disponibles. La section 2.4 se concentre sur l'ASG de modèles dynamiques fortement non linéaires construits dans l'environnement Modelica et convertis au format FMU suivant la norme FMI pour réduire les coûts de calcul. La section 2.5 décrit l'ASG réalisée sur les modèles EF. Enfin, la section 2.6 termine le chapitre par une conclusion et des perspectives d'évolution.

2.2 Analyse de sensibilité

2.2.1 Indices de Sobol et analyse de la variance

Considérons un système MIMO dont la fonction de transfert est $\mathcal{M} : \mathcal{A} \subseteq \mathbb{R}^N \rightarrow \mathcal{B} \subseteq \mathbb{R}^M$, où N est le nombre de variables d'entrée, \mathcal{A} est le domaine de définition (où les variables d'entrée prennent des valeurs), M est le nombre de réponses de sortie du

système et B est le co-domaine (où les sorties du système prennent des valeurs). $\zeta^T = (\zeta_1, \dots, \zeta_N)$, $\zeta \in \mathcal{A}$ représente le vecteur des entrées, tandis que $\mathbf{y}^T = (y_1, \dots, y_M)$, $\mathbf{y} \in \mathcal{B}$ représente le vecteur des sorties.

La variabilité des entrées est modélisée par des variables aléatoires caractérisées par leurs *Probability Density Function* (PDF) : $\zeta_i \sim dP_{\zeta_i}$, $i = 1, \dots, N$ (ce qui doit être lu : ζ_i suit une distribution de PDF dP_{ζ_i}). La version stochastique du système MIMO se lit comme suit :

$$\mathbf{y} = \mathcal{M}(\zeta), \quad \zeta \in \mathcal{A} \subseteq \mathbb{R}^N, \quad \mathbf{y} \in \mathcal{B} \subseteq \mathbb{R}^M. \quad (2.1)$$

Dans la suite de ce travail, seules les variables d'entrée indépendantes sont considérées, c'est-à-dire qu'une variable d'entrée générique ne peut pas être exprimée en tant que fonction des autres entrées (ni explicitement ni implicitement). Formellement, cela signifie que la PDF de ζ se lit comme suit :

$$dP_{\zeta} = \prod_{i=1}^N dP_{\zeta_i}. \quad (2.2)$$

Dans ce travail, les entrées du modèle sont distribuées selon une distribution uniforme \mathcal{U} dans le domaine de définition \mathcal{A} . On suppose que les résultats de l'ASG, lorsqu'on considère une distribution uniforme des variables d'entrée (à la fois continues et discrètes), ne sont pas influencés par l'étape de discrétisation. Cela signifie que les variables d'entrée considérées dans ce document sont toujours indépendantes, quelle que soit leur nature, c'est-à-dire discrètes (avec un pas défini par l'utilisateur) ou continues (qui sont nécessairement discrétisées lorsque l'on considère une distribution uniforme avec un pas de discrétisation lié à la précision de la machine). En outre dans cette section, la discussion se limite au cas des systèmes MISO, c'est-à-dire $M = 1$. Néanmoins, la formulation proposée peut être facilement étendue au cas plus général des systèmes MIMO [116].

Considérons le sous-ensemble générique $\mathcal{S} \subseteq \{1, \dots, N\}$, avec $\mathcal{S} \neq \emptyset$, ayant une cardinalité $\sharp\mathcal{S} = N_s \leq N$. Si la fonction de sortie scalaire satisfait à certaines hypothèses de base [44], la décomposition de Hoeffding [81] de la fonction de sortie y , s'écrit comme suit :

$$y = \mathcal{M}_0 + \sum_{i=1}^N \mathcal{M}_i(\zeta_i) + \sum_{1 \leq i < j \leq N} \mathcal{M}_{i,j}(\zeta_i, \zeta_j) + \dots + \mathcal{M}_{1, \dots, N}(\zeta) = \mathcal{M}_0 + \sum_{\substack{\mathbf{u} \in \mathcal{S} \\ \mathbf{u} \neq \mathbf{0}}} \mathcal{M}_{\mathbf{u}}(\zeta_{\mathbf{u}}), \quad (2.3)$$

où $\zeta_{\mathbf{u}}$ est un sous-*array* de ζ dont les indices sont définis par les composantes du vecteur \mathbf{u} dont la cardinalité est $\sharp\mathbf{u} = N_u \leq N_s$ et dont les composantes appartiennent au sous-ensemble \mathcal{S} , c'est-à-dire, $u_k \in \mathcal{S}, \overline{=1, \dots, N_u}$ [99]. L'unicité de la décomposition est assurée par les conditions suivantes :

$$\mathcal{M}_0 = \mathbb{E}(y), \quad (2.4)$$

et

$$\mathbb{E}[\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}})\mathcal{M}_{\mathbf{v}}(\boldsymbol{\zeta}_{\mathbf{v}})] = 0, \quad u_k, v_k \in \mathcal{S}, \quad \mathbf{u} \neq \mathbf{v}. \quad (2.5)$$

Dans les Eqs. (2.4) et (2.5), le symbole \mathbb{E} désigne l'espérance. Par conséquent, dans l'Eq. (2.4), \mathcal{M}_0 est la valeur moyenne. Il convient de noter que les formules ci-dessus impliquent que tous les termes $\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}})$ ($\mathbf{u} \neq \mathbf{0}$) dans l'Eq. (2.3) ont des valeurs moyennes nulles. Les termes $\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}})$ peuvent être obtenus de manière récursive comme suit :

$$\begin{cases} \mathcal{M}_i &= \mathbb{E}(y|\zeta_i) - \mathcal{M}_0, \\ \mathcal{M}_{i,j} &= \mathbb{E}(y|\zeta_i, \zeta_j) - \mathcal{M}_0 - \mathcal{M}_i - \mathcal{M}_j, \\ \dots & \end{cases} \quad (2.6)$$

Les termes d'ordre croissant dans l'Eq. (2.6) sont des espérances conditionnelles définies de manière récursive, constituant ainsi une décomposition orthogonale (et unique) de la sortie du système [81, 99].

Le but de l'ASG est de déterminer l'influence relative de chaque paramètre ζ_i sur la sortie considérée y , en termes de variance. En suivant la même idée à la base de l'Eq. (2.3), il est possible d'introduire la décomposition *ANalysis Of VAriance* (ANOVA) [44, 89, 99] comme suit :

$$\text{Var}(y) = \sum_{\substack{\mathbf{u} \in \mathcal{S} \\ \mathbf{u} \neq \mathbf{0}}} \text{Var}(\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}})). \quad (2.7)$$

Dans la formule ci-dessus, $\text{Var}(\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}}))$ représente la variance conditionnelle pour le vecteur $\boldsymbol{\zeta}_{\mathbf{u}}$ dont les indices, rassemblés dans le vecteur \mathbf{u} , appartiennent au sous-ensemble \mathcal{S} [99]. L'indice générique de Sobol $S_{\mathbf{u}}$ est défini comme suit :

$$S_{\mathbf{u}} := \frac{\text{Var}(\mathcal{M}_{\mathbf{u}}(\boldsymbol{\zeta}_{\mathbf{u}}))}{\text{Var}(y)}, \quad (2.8)$$

qui représente le rapport entre la variance due à l'interaction entre les composantes de $\boldsymbol{\zeta}_{\mathbf{u}}$ (pour $\mathbf{u} \in \mathcal{S}$) et la variance totale de la sortie. Bien entendu, les indices de Sobol satisfont à la relation suivante :

$$\sum_{\substack{\mathbf{u} \in \mathcal{S} \\ \mathbf{u} \neq \mathbf{0}}} S_{\mathbf{u}} = \sum_{i=1}^N S_i(\zeta_i) + \sum_{1 \leq i < j \leq N} S_{i,j}(\zeta_i, \zeta_j) + \dots + S_{1,\dots,N}(\boldsymbol{\zeta}) = 1. \quad (2.9)$$

Selon l'équation (2.8), l'indice de Sobol du premier ordre, également appelé indice de Sobol élémentaire, pour une seule variable ζ_i est défini comme suit :

$$S_i := \frac{\text{Var}(\mathcal{M}_i(\zeta_i))}{\text{Var}(y)}. \quad (2.10)$$

L'indice de Sobol du premier ordre fournit une mesure de l'influence de la variable d'entrée unique ζ_i sur la sortie y . Cependant, l'indice de Sobol élémentaire ne fournit aucune information sur l'influence de la variable ζ_i sur la sortie y lorsqu'elle interagit avec d'autres variables d'entrée ζ_k , $k \in \mathcal{S}$, $k \neq i$.

Pour obtenir cette information, il faut considérer les indices de Sobol d'ordre 2 à n qui, selon l'Eq. (2.8), peuvent être exprimés comme suit :

$$S_{i,j} := \frac{\text{Var}(\mathcal{M}_{i,j}(\zeta_i, \zeta_j))}{\text{Var}(y)}, S_{i,j,k} := \frac{\text{Var}(\mathcal{M}_{i,j,k}(\zeta_i, \zeta_j, \zeta_k))}{\text{Var}(y)}, \dots, \quad (2.11)$$

où $S_{i,j}$ est l'indice de Sobol de second ordre traduisant l'influence du couple (ζ_i, ζ_j) sur la sortie y , tandis que $S_{i,j,k}$ est l'indice de Sobol de troisième ordre qui fournit une mesure de l'influence des trois premières variables d'entrée $(\zeta_i, \zeta_j, \zeta_k)$ sur la sortie y , etc.

Les $2^n - 1$ indices de Sobol peuvent fournir des informations précieuses pour l'ASG, mais leur calcul est prohibitif lorsqu'un grand nombre de variables est considéré. À cette fin, une mesure souvent appelée « indice d'effet total » ou « indice d'ordre total » ou encore « indice de Sobol total », noté S_{T_i} , est utilisée [99, 151]. L'indice de Sobol total lié à la variable d'entrée ζ_i peut être défini comme [99, 152] :

$$S_{T_i} := \sum_{\substack{\mathbf{u} \in \mathcal{S} \\ i \in \mathbf{u} \neq \mathbf{0}}} S_{\mathbf{u}}. \quad (2.12)$$

L'indice de Sobol total S_{T_i} représente l'effet total de la variable d'entrée générique ζ_i , comprenant son effet direct sur la sortie ainsi que toutes les interactions avec d'autres variables d'entrée. Selon l'Eq. (2.12), il est égal à la somme de tous les indices partiels $S_{\mathbf{u}}$ impliquant la variable d'entrée ζ_i . Bien entendu, contrairement aux indices élémentaires S_i , la somme de tous les indices de Sobol totaux peut être supérieure ou égale à un, i.e.,

$$\sum_{i=1}^n S_{T_i} \geq 1. \quad (2.13)$$

Cela est dû au fait que l'interaction entre deux variables, par exemple ζ_i et ζ_j , est comptabilisée dans les deux indices totaux associés, c'est-à-dire S_{T_i} et S_{T_j} . La somme des indices totaux est égale à un uniquement lorsque le modèle est purement additif. Il convient de noter que dans [151], Saltelli a présenté un algorithme nécessitant $n(N+2)$ simulations, où n est le nombre total d'échantillons permettant le calcul des N indices élémentaires $(S_i)_{i \in [1, N]}$ ainsi que des N indices totaux. D'après l'Eq. (2.12), si $S_{T_i} \approx 0$, on peut affirmer que la variable ζ_i n'influence pas du tout la sortie considérée.

Dans la suite de ce chapitre (et, d'une manière plus générale, dans la plupart des problèmes d'ingénierie), seuls les indices de Sobol élémentaires et totaux sont pris en compte. Les indices de Sobol élémentaires sont désignés par S_i , tandis que les indices de Sobol totaux sont désignés par S_{T_i} . Pour avoir une vision plus approfondie de la question,

le lecteur intéressé pourra consulter [22, 30, 89, 90, 99].

2.2.1.1 Algorithme de Saltelli

Les algorithmes de Saltelli présentés dans [151, 152] pour évaluer les indices de Sobol nécessitent le même nombre de simulations, c'est-à-dire $n(N + 2)$ avec n le nombre d'échantillons, et fournissent les indices élémentaires et totaux, c'est-à-dire S_i et S_{T_i} , respectivement. Ils diffèrent essentiellement par la manière dont ils calculent ces indices. Plus précisément, dans [152], la technique Quasi Monte-Carlo (QMC) est utilisée pour générer les échantillons. Pour un aperçu plus approfondi de la question, le lecteur intéressé est invité à consulter [151, 152].

2.2.1.2 Méthode *Fourier Amplitude Sensitivity Test*

La méthode FAST utilise l'échantillonnage périodique et la transformation de Fourier pour décomposer la variance d'une sortie de modèle en variances partielles liées aux différents paramètres du modèle. En reliant ces variances partielles à la variance globale, il est possible de calculer les indices de Sobol.

Il convient de noter que la méthode FAST est une technique relativement ancienne [42], mais qu'elle a été améliorée au cours de la dernière décennie en combinant la technique des plans *Random Balance Design* (RBD) avec l'algorithme FAST, i.e., l'algorithme RBD-FAST [67, 140, 175, 177]. Cependant, la plupart de ces améliorations sont soit indisponibles dans les bibliothèques classiques de Python ou de MATLAB[®], soit incomplètes. Par exemple, la méthode RBD-FAST, intégrée dans SALib¹, ne calcule que les indices du premier ordre, bien que cette méthode ait été étendue aux indices totaux [117]. Le lecteur intéressé est invité à consulter [43, 153] pour plus de détails sur le contexte théorique de la méthode FAST.

L'un des avantages de cette méthode par rapport aux algorithmes de Saltelli est qu'elle nécessite un nombre d'échantillon n et plus globalement un nombre de simulations plus petit pour calculer les indices de Sobol. En effet, nN simulations sont nécessaires contre $n(N + 2)$ pour les algorithmes de Saltelli. Néanmoins, certaines précautions doivent être prises lors de l'utilisation de cette méthode. Tissot et Prieur [177] ont montré que pour des modèles comportant plus de 10 variables d'entrée, les valeurs des indices étaient biaisées et l'algorithme devenait instable.

Contrairement aux algorithmes de Saltelli qui sont basés sur la décomposition ANOVA, la méthode FAST utilise la décomposition en fréquence des sorties du système. Cela entraîne de légères différences entre les valeurs des indices de Sobol et celles résultant de la méthode FAST. Néanmoins, ces différences sont négligeables (si le problème comporte moins de 10 variables) et les informations fournies par l'algorithme FAST peuvent être

1. <https://salib.readthedocs.io/en/latest/>

utilisées pour quantifier et classer les influences des différentes variables d'entrée.

2.2.2 Indices de Shapley

Récemment, Owen [135] a proposé une méthode d'ASG pour les systèmes caractérisés par des variables d'entrée dépendantes, basée sur l'effet de Shapley. Il s'agit d'un concept tiré de la théorie des jeux [159] visant à évaluer la contribution de chaque joueur à remporter un gain commun dans un jeu coopératif. Néanmoins, seuls les problèmes dynamiques non linéaires caractérisés par des variables d'entrée indépendantes sont considérés dans ce travail. Ainsi, seule la formulation des indices de Shapley pour cette famille de problèmes est brièvement rappelée ci-dessous.

Selon [79, 90], et en utilisant la notation introduite dans la formulation des indices de Sobol, dans le cas de variables d'entrée indépendantes, l'indice de Shapley lié à l'entrée générique ζ_i est défini comme suit :

$$S_{H_i} := \sum_{\substack{\mathbf{u} \in \mathcal{S} \\ i \in \mathbf{u} \neq \mathbf{0}}} \frac{S_{\mathbf{u}}}{N_{\mathbf{u}}}, \quad (2.14)$$

où $N_{\mathbf{u}}$ est la cardinalité de la matrice \mathbf{u} rassemblant les indices $u_k \in \mathcal{S}$ utilisés pour calculer l'indice élémentaire $S_{\mathbf{u}}$, c'est-à-dire, $N_{\mathbf{u}} = 1$ si $\mathbf{u}^T = (1)$, $N_{\mathbf{u}} = 2$ si $\mathbf{u}^T = (1, 2)$, etc.

Il convient de noter que, selon l'Eq. (2.14), l'indice de Shapley lié à la variable d'entrée générique ζ_i est supérieur ou égal à l'indice de Sobol élémentaire correspondant et inférieur ou égal à l'indice de Sobol total correspondant, c'est-à-dire $S_i \leq S_{H_i} \leq S_{T_i}$. Bien sûr, cette propriété est due à la définition mathématique de l'indice de Shapley, mais elle suit également une logique claire : l'indice de Sobol du premier ordre quantifie l'influence d'une seule variable d'entrée sur la sortie, l'indice de Sobol total prend en compte l'influence de l'entrée unique sur la sortie ainsi que les interactions avec les autres variables d'entrée appartenant à $\zeta_{\mathbf{u}}$ (et les interactions sont évaluées plusieurs fois), tandis que dans la définition de l'indice de Shapley, l'effet d'interaction est réparti de manière égale entre chaque variable d'entrée impliquée dans l'interaction. Dans la suite de ce travail, les indices de Shapley sont désignés par S_{H_i} .

Un aperçu des principales caractéristiques des indices de Sobol et de Shapley ainsi que les algorithmes utilisés dans cette thèse sont présentés dans le Tab. 2.1.

2.3 Problème analytique de référence : la fonction d'Ishigami

La fonction d'Ishigami [15, 91] est largement utilisée comme fonction test de référence dans l'ASG en raison de son comportement fortement non linéaire et non monotone [16, 189]. En outre, sa dépendance à l'égard de la troisième variable est assez particulière,

TABLE 2.1 – Principales caractéristiques des indices de Sobol, des indices de Shapley et des algorithmes connexes utilisés dans cette étude.

Sobol		
Indice élémentaire		Indice totaux
<ul style="list-style-type: none"> - Influence d’un paramètre - $0 \leq S_i \leq 1$ 		<ul style="list-style-type: none"> - Influence d’un paramètre et de ses interactions - Compte les interactions plusieurs fois - $\sum_{i=1}^N S_{T_i} \geq 1$
Algorithme de Saltelli 2002 [151]	Algorithme de Saltelli 2010 [152]	Algorithme de FAST [42, 153]
<ul style="list-style-type: none"> - Échantillonnage MC - $n(N+2)$ échantillons - $S_i, S_{i,j}, S_{T_i}$ - Décomposition ANOVA 	<ul style="list-style-type: none"> - Échantillonnage QMC selon l’algorithme de Sobol - $n(N+2)$ échantillons - $S_i, S_{i,j}, S_{T_i}$ - Décomposition ANOVA - Formules pour déterminer les indices d’une manière plus efficace que dans [151] 	<ul style="list-style-type: none"> - Échantillonnage QMC selon l’algorithme de Sobol - nN échantillons - S_i, S_{T_i} - Décomposition fréquentielle
Effet Shapley		
<ul style="list-style-type: none"> - Influence d’un paramètre et de sa contribution dans chacune de ses interactions - $\sum_{i=1}^N S_{H_i} = 1$ - $0 \leq S_{H_i} \leq 1$ - $S_i \leq S_{H_i} \leq S_{T_i}$ 		
Algorithme de Goda 2021 [66]		
<ul style="list-style-type: none"> - Échantillonnage QMC selon l’algorithme de Sobol - $n(N+1)$ échantillons - S_{H_i} 		

comme le décrit [165]. Elle s’exprime par :

$$f(\mathbf{x}) = \sin(x_1) + a \sin^2(x_2) + b x_3^4 x_1, \tag{2.15}$$

avec $x_i \in \mathcal{U}[-\pi; \pi], i = 1, 2, 3, a = 7$ et $b = 0.1$. Les valeurs des paramètres a et b sont tirées de [189] pour pouvoir utiliser par la suite les résultats de ce même article comme référence.

Pour calculer les indices de Sobol selon l’algorithme de Saltelli 2002 [151], la bibliothèque OpenTURNS est utilisée, tandis que pour l’algorithme de Saltelli 2010 [152] et l’algorithme FAST [153], c’est la bibliothèque SALib qui est employée. Les deux bibliothèques sont disponibles en Python. Pour les indices de Shapley, l’algorithme développé

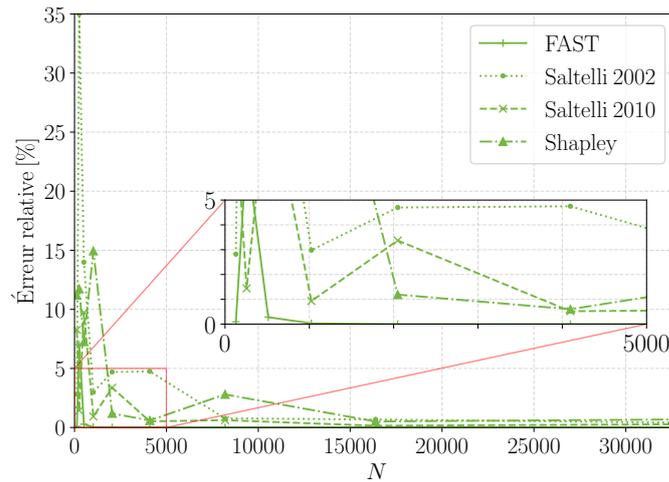


FIGURE 2.1 – Erreur relative de l'indice élémentaire de Sobol et de l'indice de Shapley associés à la variable x_1 en fonction du nombre d'échantillons. Ces indices sont calculés avec les algorithmes de Saltelli de 2002 [151] et de 2010 [152], l'algorithme FAST ainsi que l'algorithme proposé par Goda [66] pour comparer les convergences de chacun.

par Goda [66] adapté au langage Python est utilisé (voir l'annexe D.1). Ces algorithmes sont comparés selon les critères suivants :

- Le taux de convergence de l'erreur relative entre l'indice de Sobol (élémentaire ou total) et l'indice de Shapley calculés par les algorithmes considérés et une valeur de référence calculée par les mêmes algorithmes lorsqu'on considère un grand nombre d'échantillons, c'est-à-dire 2^{22} . L'erreur relative sur chaque indice est acceptable lorsqu'elle est inférieure ou égale à 1 % pour 10^4 échantillons successifs. Bien entendu, plus le nombre d'échantillons pour atteindre la convergence est faible, plus l'algorithme est efficace.
- La précision de l'algorithme dans l'évaluation des indices de Sobol et de Shapley, lorsque la convergence est atteinte, en comparant les valeurs fournies par l'algorithme considéré aux valeurs analytiques disponibles dans la littérature [16, 189].
- L'Intervalle de Confiance (IC) pour chaque indice avec un niveau de confiance égale à 95 %, qui fournit une mesure de la fiabilité de l'algorithme dans l'évaluation des indices considérés.

Le taux de convergence de l'erreur relative sur l'indice élémentaire et sur l'indice de Shapley pour la variable d'entrée x_1 est illustré dans la Fig. 2.1 (les résultats relatifs aux variables x_2 et x_3 sont caractérisés par la même tendance mais ne sont pas illustrés par souci de concision). Comme le montre cette figure, la méthode FAST nécessite un nombre d'échantillons inférieur à celui requis par les algorithmes de Saltelli pour atteindre la convergence. Quant à la convergence de l'indice de Shapley, elle est plus lente. Néanmoins, ces indices seront considérés dans la suite du manuscrit, car ils complètent les informations fournies par les indices de Sobol en se positionnant comme résultats intermédiaire entre

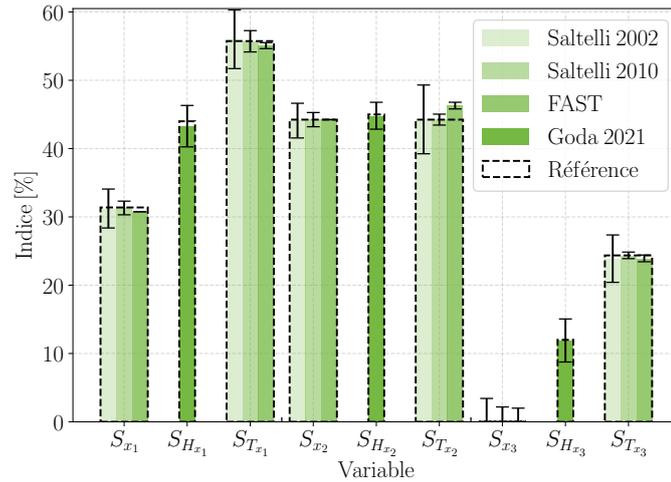


FIGURE 2.2 – Comparaison entre les indices élémentaires et totaux de Sobol et les indices de Shapley avec leur valeur analytique sur la fonction d'Ishigami [16, 189]. L'Intervalle de Confiance à 95 % est également représenté.

les indices élémentaires et totaux.

La Fig. 2.2 représente les valeurs des indices de Sobol et de Shapley calculés par les différents algorithmes étudiés et comparés aux valeurs de références tirées de [16, 189]. Les valeurs retenues sont évaluées sur un nombre d'échantillon de 2^{15} , quantité pour laquelle tous les algorithmes convergent de manière certaine. La Fig. 2.2 fournit également l'IC à 95 % pour chaque indices calculés par chaque.

L'analyse de ce graphique permet de conclure que les résultats fournis par l'algorithme de Saltelli 2010 [152] sont les plus précis avec une erreur relative maximale sur l'indice total associé à la variable x_3 de 1.98 %. En revanche, les résultats ayant le moins d'incertitude, c'est-à-dire les plus fiables, sont ceux fournis par la méthode FAST [42, 153] avec un IC autour de la valeur de l'indice total de la variable x_3 de 0.49 %. Il convient de noter que l'algorithme FAST à une bonne précision, avec le pourcentage d'erreur le plus élevé égal à 4,65 % pour l'indice total lié à la variable x_2 . En outre, l'indice de Shapley prend des valeurs comprises entre l'indice élémentaire et l'indice total pour chaque variable et comporte un IC moyen de 2.71 % et une erreur relative moyenne de 0.94 %.

Par conséquent, pour les autres problèmes considérés dans ce document, l'algorithme FAST a été retenu pour évaluer les indices de Sobol, car il représente le meilleur compromis entre précision, fiabilité et coûts de calcul : les résultats calculés sur un nombre d'échantillon de 2^{15} sont les mêmes que ceux évalués sur un nombre d'échantillon de 2^{12} . Dans ce qui suit, la méthode FAST sera donc utilisée avec un nombre d'échantillons égal à 2^{12} , comparée aux solutions de référence de la littérature ou aux valeurs calculées par l'algorithme de Saltelli 2010 avec une taille d'échantillon de 2^{22} . En outre, l'algorithme proposé par Goda [66] sera utilisé pour calculer les indices de Shapley, avec un nombre d'échantillons de 2^{14} .

2.4 Problèmes non-linéaires dynamiques - Approche MBSE

Cette section a pour but, d'illustrer l'emploi d'une ASG sur des modèles MBSE non-linéaires. Le premier, un amortisseur de choc utilisé dans une voiture, est un cas non-linéaire où l'objectif est de déterminer les influences de ses variables de conception sur son état à l'équilibre. Une étude de l'évolution de l'influence de ces variables est également faite durant sa phase transitoire. Le second cas, est un benchmark (*Stick-Slip*) répandu dans la littérature pour son aspect temporel fortement non-linéaire et sa difficulté de résolution par des langages classiques tel que FORTRAN. L'intérêt mis en avant ici est l'efficacité de l'approche MBSE pour modéliser ce type de système, indépendant de tout problème numérique pour l'utilisateur. Le but est ensuite d'appliquer une ASG pour quantifier l'influence de ses variables de conception sur sa stabilité. Un aspect remarquable de ce second cas test est que sa sortie d'intérêt est un booléen. Celui-ci qualifie l'atteinte d'un régime stable ou non, contrairement au cas de l'amortisseur où la sortie est une variable continue qui représente sa position finale.

2.4.1 L'amortisseur de chocs

2.4.1.1 Description analytique

Le premier système mécanique analysé dans ce chapitre est un amortisseur de voiture, illustré par son schéma fonctionnel à la Fig. 2.3. Quatre éléments principaux peuvent être identifiés :

- un piston avec un trou central qui permet l'écoulement de l'huile pendant le mouvement. Cela permet également d'absorber les oscillations du ressort. Le rayon externe du piston est égal au rayon interne de la chambre du piston, sans jeux ;
- un ressort en parallèle du piston ;
- les masses des passagers et de la voiture appliquées au support supérieur de l'amortisseur ;
- la roue en contact direct avec le sol fixée au support inférieur de l'amortisseur.

La réponse dynamique de ce système est régie par le déplacement longitudinal $x(t)$ du piston de la chambre supérieure. Il peut être déterminé en résolvant l'équation d'équilibre du système :

$$M\mathbf{a} = \sum_{i=1}^q \mathbf{f}_{\text{ext},i}, \quad (2.16)$$

où $M = \frac{m_P}{4} + \frac{m_C}{4}$ est le quart de la masse totale du système, comprenant celles de la voiture et des passagers, \mathbf{a} est l'accélération, dont la composante le long de l'axe x est égale à $\ddot{x}(t)$, $\mathbf{f}_{\text{ext},i}$ est la i -ième force externe générique appliquée au ressort-amortisseur

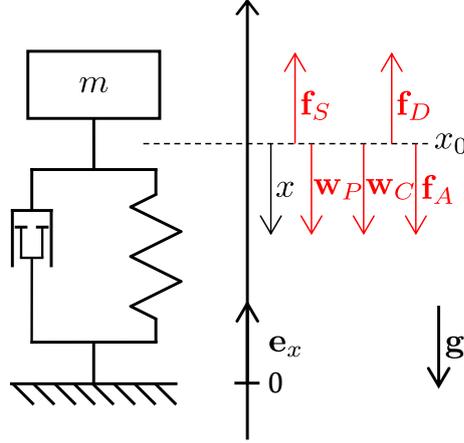


FIGURE 2.3 – Schéma fonctionnel de l'amortisseur de voiture.

(voir Fig. 2.3) et q est le nombre total de forces appliquées.

Les forces externes appliquées au système sont énumérées ci-dessous :

— poussée d'Archimède et force du ressort :

$$\mathbf{f}_A = -S_p h \rho_{\text{hu}} g \mathbf{e}_x, \quad \mathbf{f}_S = -k(x - L_0) \mathbf{e}_x, \quad (2.17)$$

où $S_p = \pi(r_{\text{ext}}^2 - r_{\text{int}}^2)$ est la section transversale du piston, r_{int} et r_{ext} sont le rayon intérieur et le rayon extérieur du piston, respectivement, h est la hauteur du piston, ρ_{hu} est la densité de l'huile, g est l'accélération gravitationnelle et \mathbf{e}_x est le vecteur unitaire de l'axe x . En ce qui concerne la force du ressort, k est la raideur du ressort et L_0 est sa longueur libre. A $t = 0$ s, le ressort est précontraint.

— force d'amortissement

$$\mathbf{f}_D = -\text{sign}(\dot{x}) S_p \Delta P \mathbf{e}_x, \quad (2.18)$$

où ΔP est la perte de charge, calculée en considérant un piston creux glissant sans jeu dans sa chambre, i.e. :

$$\Delta P = \rho_{\text{hu}} \frac{\omega_0^2}{2} (\zeta_{\text{red}} + \zeta_{\text{exp}}), \quad \omega_0 = \frac{\dot{x} S_p}{S_h}, \quad (2.19)$$

où $S_h = \pi r_{\text{int}}^2$ est la section transversale du trou du piston. En outre, les paramètres ζ_{red} et ζ_{exp} sont exprimés comme suit :

$$\zeta_{\text{red}} = \left(\frac{1}{C} - 1 \right)^2, \quad \zeta_{\text{exp}} = \left(1 - \frac{S_h}{S_p} \right)^2, \quad (2.20)$$

avec $C = 0.63 + 0.37 \left(\frac{S_h}{S_p} \right)^2$.

ment, à une situation physiquement admissible. Notons que la position initiale du piston est définie comme $x(0) = L_{\text{bar}} + L_{\text{cham}}$, où L_{bar} est la distance avec le sol et L_{cham} est la longueur de la chambre du piston, qui limite son déplacement. A $t = 0$ s, le ressort est précontraint : $x(0) < L_0 + L_{\text{bar}}$.

TABLE 2.2 – Valeurs de référence et intervalles de variation.

Variable	Référence	Intervalle
r_{ext} [m]	0.03	[–]
L_{cham} [m]	0.25	[–]
L_{bar} [m]	0.5	[–]
ρ_{hu}	884	[–]
g [ms^{-2}]	9.81	[–]
k [Nm^{-1}]	9000	$\mathcal{U}([0.02, 0.04])$
L_0 [m]	0.45	$\mathcal{U}([0.4, 0.6])$
r_{int} [m]	0.002	$\mathcal{U}([0.001, 0.005])$
m_P [kg]	150	$\mathcal{U}([0.01, 0.04])$
m_C [kg]	1000	$\mathcal{U}([900, 1300])$
h [m]	0.02	$\mathcal{U}([60, 360])$

Une fois le modèle Modelica créé et validé, le fichier FMU est généré en mode co-simulation, c'est-à-dire qu'il est complètement autonome grâce au solveur temporel qu'il comporte. La limite de cet exercice est que, selon le cas d'étude, le solveur temporel (ici de type Runge-Kutta 4) peut être moins performant que celui utilisé dans le logiciel MBSE initial. Pendant l'ASG, le FMU est considérée comme une boîte noire dont les variables d'entrée sont échantillonnées suivant une loi uniforme \mathcal{U} , dans leurs intervalles respectifs, comme indiqué dans le Tab. 2.2. Ces intervalles ont été choisis en fonction des conditions mécaniques admissibles des amortisseurs. En particulier, l'intervalle de définition du paramètre r_{int} est choisi pour garantir un temps caractéristique d'amortissement complet ou semi-périodique.

Lors de la génération du fichier FMU en mode co-simulation, il a été nécessaire de définir les entrées, les sorties et les paramètres modifiables du modèle, mais aussi le type de solveur intégré dans le fichier. Dans cette situation, un algorithme de Runge-Kutta du quatrième ordre a été choisi, dont les pas de temps d'intégration et de communication avec le programme global sont fixés à 10^{-5} s. Compte tenu de ces caractéristiques, le temps de simulation du modèle de référence de l'amortisseur est réduit d'un facteur ≈ 100 en passant du logiciel MapleSim initial à un appel du fichier FMU (code C) par Python, adéquat pour mener à terme une étude statistique telle qu'une ASG.

Remarque 1 *Avant de passer à l'ASG, le modèle étudié peut être simplifié. Lors de l'évaluation des ordres de grandeur des forces externes, la force d'Archimède est négligeable devant la force du ressort : la première est environ 10^3 fois plus petite que la seconde. Par conséquent, le paramètre h peut être supprimé des variables d'entrée considérées dans l'ASG.*

2.4.1.3 Résultats numériques

En considérant l'intervalle de variation de chaque variable d'entrée, le mouvement du piston peut être de trois types : immobile à sa position initiale (dans ce cas la pré-charge due au ressort est dominante), complètement amorti ou semi-périodique. La réponse de sortie étant la position finale du piston, une ASG a été réalisée en observant $x(t)$ pour dix temps caractéristiques, soit $t = i\tau_{\text{abs}}$, avec $i \in [1, \dots, 10]$ et $\tau_{\text{abs}} = 2$ s. Les bornes de cet intervalle correspondent à un temps intermédiaire du comportement lorsque $i = 1$, et à un temps suffisamment élevé pour atteindre la position stable lorsque $i = 10$.

Les résultats de l'ASG sont présentés dans la Fig. 2.5 pour différentes valeurs de t dans l'intervalle $[\tau_{\text{abs}}, 10\tau_{\text{abs}}]$. Pour ces analyses, la méthode FAST a été utilisée pour évaluer les indices de Sobol et la version Python de l'algorithme proposé par Goda [66] pour évaluer les indices de Shapley. En outre, lorsque $t = 10\tau_{\text{abs}}$ les indices sont comparés aux résultats de référence obtenus par l'algorithme de Saltelli 2010 avec $N = 2^{22}$ échantillons.

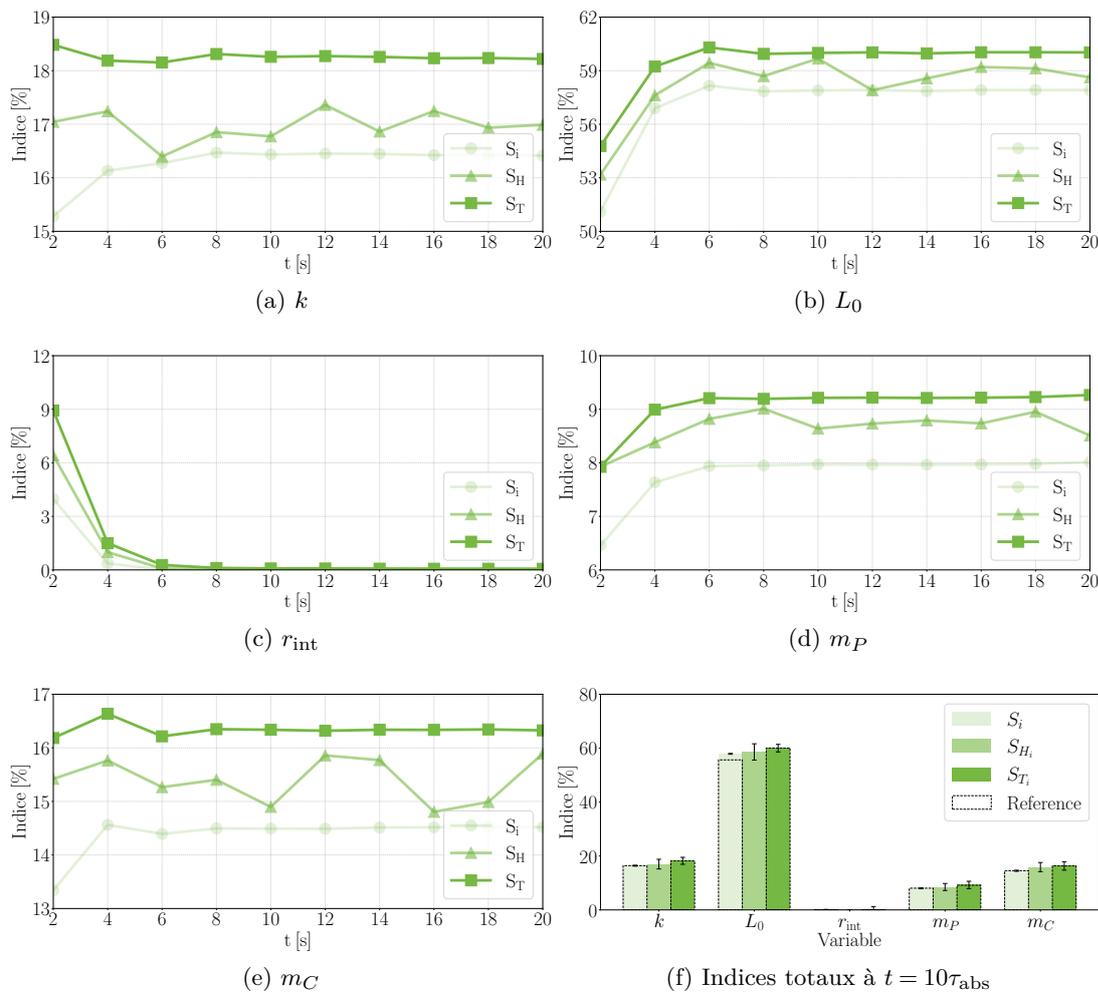


FIGURE 2.5 – Évolution des indices de Sobol et de Shapley élémentaires et globaux au cours du temps pour les variables (a) k , (b) L_0 , (c) r_{int} , (d) m_P , (e) m_C . En (f), la comparaison entre les indices globaux et la solution de référence est illustrée à $t = 10\tau_{\text{abs}}$.

Comme on peut le déduire de la Fig. 2.5, les indices de Sobol totaux des variables r_{int} et m_P sont inférieurs à ceux des autres variables d'entrée à $t = 10\tau_{\text{abs}}$. Néanmoins, l'influence de la variable r_{int} diminue avec le temps et devient nulle lorsque $t > 3\tau_{\text{abs}}$, tandis que celle de m_P augmente. C'est un résultat cohérent compte tenu que r_{int} intervient seulement dans l'expression de la force d'amortissement. Cette force dépend de la vitesse du piston, ainsi, lorsque le système atteint un état stable, la vitesse du piston devient nulle et l'influence de la variable r_{int} devient nulle également.

D'après la définition des indices élémentaires et totaux des Eqs. (2.10) et (2.12), respectivement, et en considérant les courbes de la Fig. 2.5, on peut conclure que, pour chaque variable d'entrée, l'influence de l'interaction avec le reste des variables sur la sortie est relativement faible. Elle varie entre 3.2% à $t = \tau_{\text{abs}}$ et 1.8% à $t = 10\tau_{\text{abs}}$ pour la variable k , 3.7% à $t = \tau_{\text{abs}}$ et 2.1% à $t = 10\tau_{\text{abs}}$ pour la variable L_0 , 5.0% à $t = \tau_{\text{abs}}$ et 0.006% à $t = 10\tau_{\text{abs}}$ pour la variable r_{int} , 1.5% à $t = \tau_{\text{abs}}$ et 1.3% à $t = 10\tau_{\text{abs}}$ pour la variable m_P , 2.8% à $t = \tau_{\text{abs}}$ et 1.8% à $t = 10\tau_{\text{abs}}$ pour la variable m_C . De plus, en classant les indices totaux de Sobol par ordre croissant, on peut conclure que le paramètre L_0 est celui qui influence le plus le mouvement du piston pour toutes les valeurs du temps caractéristique.

Comme on peut le déduire des résultats ci-dessus, r_{int} a un rôle mineur sur le comportement du système pour $t > 3\tau_{\text{abs}}$. Cependant, si l'on considère une plage de variation plus large, avec des valeurs de r_{int} supérieures à 0,005 m par exemple, le piston est caractérisé par un mouvement instable dans la plupart des cas. Cela souligne l'importance de définir correctement les intervalles de définition des variables d'entrée pour mener une ASG. En conséquence, r_{int} est la variable d'entrée ayant le moins d'influence sur le mouvement du piston lorsqu'il est question de sa position à l'équilibre.

L'indice de Shapley lié à chaque variable d'entrée a été évalué dans les mêmes conditions que celles considérées pour l'ASG basée sur les indices de Sobol. Ils sont également reportés dans la Fig. 2.5. Leurs valeurs sont toujours comprises entre S_i et S_{T_i} , pour chaque temps, et leur somme est toujours égale à un. Globalement, ils fournissent la même information que S_{T_i} , bien que leur comportement soit bien moins stable et donc bien moins fiable. En ce qui concerne la fiabilité des indices calculés, comme le montre la Fig. 2.5f à $t = 10\tau_{\text{abs}}$, la dispersion des indices de Shapley est plus élevée que celle des indices de Sobol, en accord avec les résultats obtenus pour la fonction d'Ishigami.

Conclusion intermédiaire 1 *Ce premier cas test est un problème non-linéaire classique d'oscillateur. L'ASG réalisée dessus a mis en évidence l'importance de bien définir le cadre de l'étude. En l'occurrence, selon la sortie observée et les plages de variation des différentes variables, les résultats peuvent être très différents. Par exemple, le rayon interne du piston a un rôle seulement dans la phase d'amortissement.*

2.4.2 Oscillateur à frottement sec

2.4.2.1 Description analytique

Le deuxième système dynamique étudié dans ce travail est un oscillateur à frottement sec. Il est modélisé à l'aide d'un système masse-ressort soumis à une force de frottement provenant d'un tapis roulant, comme le montre la Fig. 2.6. Ce système a été largement étudié dans [111,161] en tant que problème de référence en raison de la forte non-linéarité introduite par la force de frottement (seuil discontinu entre friction et glissement).

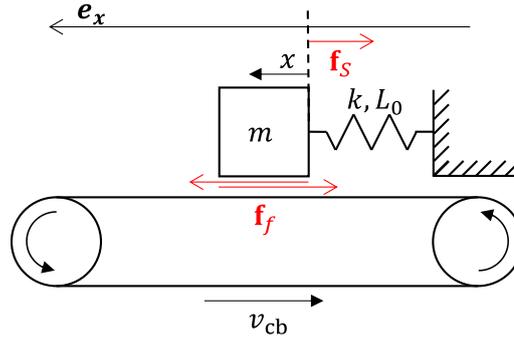


FIGURE 2.6 – Représentation schématique de l'oscillateur à friction sèche.

L'équation du mouvement de ce système suivant l'axe e_x est la suivante :

$$m\mathbf{a} = \sum_{i=1}^q \mathbf{f}_{\text{ext},i}, \quad (2.22)$$

où m est la masse, \mathbf{a} est l'accélération (dont la composante le long de l'axe x est désignée par $\ddot{x}(t)$), $\mathbf{f}_{\text{ext},i}$ est la i -ième force externe appliquée à la masse et q est le nombre total de forces externes.

Les forces externes appliquées à la masse sont énumérées ci-dessous :

— force du ressort :

$$\mathbf{f}_S = -k(x - L_0)\mathbf{e}_x, \quad (2.23)$$

où k est la constante du ressort et L_0 sa longueur libre.

— force de friction :

$$\mathbf{f}_F = -\frac{F_s \text{sign}(v_{\text{rel}})}{1 + \delta|v_{\text{rel}}|} \mathbf{e}_x, \quad (2.24)$$

où $F_s = 1$ N est la valeur seuil du frottement, $v_{\text{rel}} = \dot{x} - v_{cb}$ est la composante selon l'axe x de la vitesse relative de la masse par rapport au tapis roulant, \dot{x} est la vitesse de la masse et v_{cb} est la vitesse du tapis roulant, qui est constante, et δ est une constante. Le modèle de la force de frottement fournit par l'Eq. (2.24) est tiré de [111].

2.4.2.2 Modèle MBSE

Comme expliqué dans [111], ce modèle se caractérise par des problèmes numériques lors de la résolution de l'équation du mouvement. Cela est dû au profil non linéaire (hyperbole) de la force de frottement \mathbf{f}_F en fonction de la vitesse relative de la masse. Pour surmonter ces problèmes, deux stratégies numériques sont généralement employées : la méthode de lissage et le modèle de commutation.

La méthode de lissage consiste à introduire un terme qui rend la courbe continue pour $1 + \delta|v_{\text{rel}}| = 0$. A l'inverse, le modèle de commutation est une succession de tests conditionnels permettant de déterminer le mode de comportement du système (bloqué, glissant ou en transition) afin d'appliquer la bonne description analytique.

Cependant, les spécificités du langage de programmation Modelica (i.e., la stratégie d'assemblage de blocs élémentaires couplée à l'aspect acausale) permettent de s'affranchir de ces complications et d'introduire directement l'équation de la force de frottement, sans aucune approximation, dans un bloc adapté. Le modèle résultant est illustré dans la Fig. 2.7 et les résultats obtenus en considérant les paramètres de référence énumérés dans Tab. 2.3 coïncident avec ceux présentés dans [111].

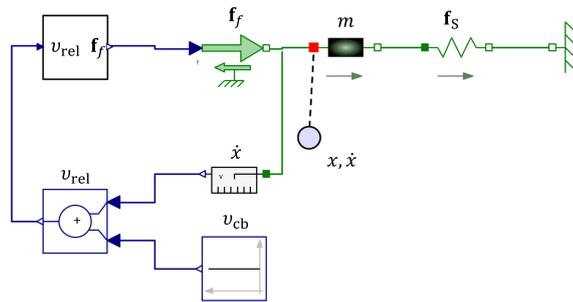


FIGURE 2.7 – Modèle *Model-Based System Engineering* du système d'oscillateur à friction sèche.

Ces résultats sont illustrés par le diagramme de phase de la Fig. 2.8 qui montre l'évolution de la vitesse de la masse en fonction de sa position. Deux régimes peuvent être observés : un régime transitoire en pointillé qui évolue vers un régime stable en trait plein. Un plateau apparaît lorsque la masse a une vitesse relative nulle par rapport au tapis roulant.

Pour cet exemple, le fichier FMU est exporté en mode de co-simulation, en incorporant un solveur basé sur l'algorithme de Runge-Kutta du quatrième ordre avec des pas de temps d'intégration et de communication tous deux égaux à 10^{-5} s.

2.4.2.3 Résultats numériques

Avant de commencer l'ASG, il faut définir l'observable. Dans le cas de l'oscillateur à frottement sec, il s'agit du comportement de la masse, stable ou instable. Plus précisément,

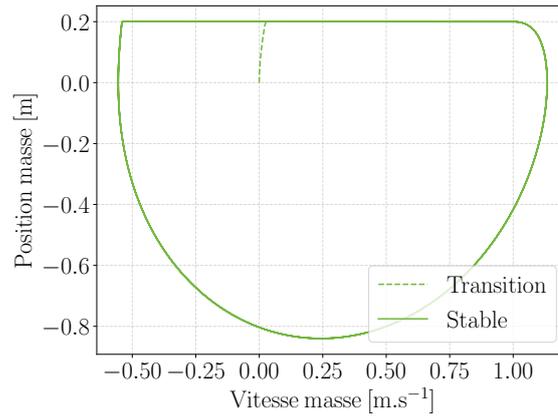


FIGURE 2.8 – Diagramme de phase de référence représentant l'évolution de la vitesse de la masse (en m.s^{-1}) en fonction de sa position (en m), obtenu avec Modelica.

le comportement est considéré comme stable lorsque la vitesse relative de la masse est nulle (adhérence) pendant un intervalle de temps Δt_{stick} supérieur ou égal à $\tau_{\text{stable}} = 0,3$ s. Inversement, lorsque la vitesse de masse n'a pas de partie constante ou est nulle pendant un intervalle de temps $\Delta t_{\text{stick}} < \tau_{\text{stable}}$, le comportement est considéré comme instable. Un cas de comportement stable est illustré dans la Fig. 2.8. Par conséquent, la réponse de sortie est l'intervalle de temps Δt_{stick} qui doit être supérieur à τ_{stable} lorsqu'un comportement stable est atteint.

TABLE 2.3 – Variables constantes et variables d'entrée : valeurs de référence et intervalles de variation.

Variable	Référence	Intervalle
k [Nm^{-1}]	1.0	$\mathcal{U}([0.5, 5.0])$
L_0 [m]	0.0	$\mathcal{U}([0.2, 1.2])$
m [kg]	1.0	$\mathcal{U}([0.5, 2.0])$
v_{cb} [ms^{-1}]	0.2	$\mathcal{U}([0.2, 5.0])$
F_S [N]	1.0	$\mathcal{U}([1.0, 5.0])$
δ [sm^{-1}]	3.0	constant

Les paramètres utilisés comme variables d'entrée, avec les plages de définition correspondantes, sont énumérés dans le Tab. 2.3. Comme dans le cas de l'amortisseur de voiture, les plages de variation ont été sélectionnées pour conserver un comportement physiquement acceptable du système.

Les résultats de l'analyse de Sobol selon la méthode FAST sont présentés dans la Fig. 2.9 et comparés aux résultats de référence obtenus avec la méthode Saltelli 2010 pour $N = 2^{22}$. De l'analyse de ces résultats, on peut déduire que l'influence de l'interaction entre les variables est significative en raison de la grande différence entre les indices totaux et élémentaires. En effet, l'influence de l'interaction est de l'ordre de 48 %.

En outre, d'après l'histogramme illustré à la Fig. 2.9, on peut déduire que la variable d'entrée ayant la plus forte influence sur la stabilité du mouvement est v_{cb} avec des valeurs

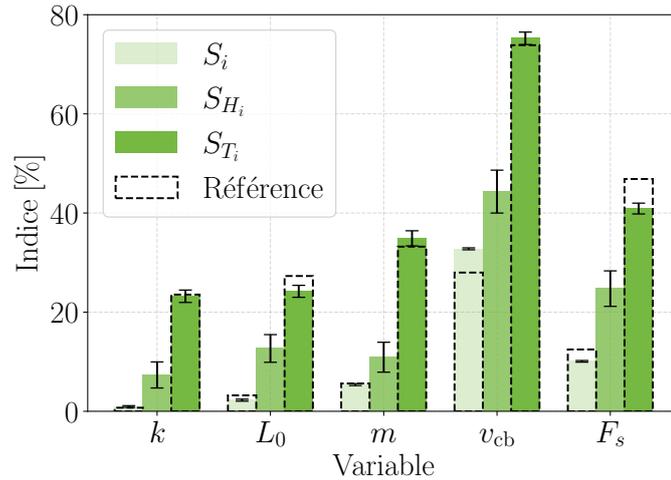


FIGURE 2.9 – Résultats de l’ASG sur le modèle d’oscillateur à friction sèche.

de S_i et S_{T_i} égales à 32 % et 75 %, tandis que celle ayant l’influence la plus faible est k avec des indices élémentaires et totaux égaux à 1 % et 23 %, respectivement. Cependant, bien que l’indice élémentaire de k soit négligeable par rapport à celui des autres variables, la valeur de l’indice total ne peut être négligée. Cela montre que cette variable influence la stabilité du mouvement lorsqu’elle est combinée avec les autres entrées. En conséquence, une attention particulière doit être accordée au choix du modèle de force de frottement dans lequel la variable v_{cb} est impliquée.

Cette conclusion se retrouve dans la littérature puisque le même modèle est utilisé avec différentes expressions de la force de frottement afin de mettre en évidence des comportements particuliers (la bifurcation de Hopf ou la bifurcation de glissement dans [161]). Contrairement aux valeurs obtenues pour les indices de Sobol dans le cas de l’amortisseur de voiture, la précision de l’évaluation des indices totaux des variables L_0 et F_s avec l’algorithme FAST n’est pas suffisante, le pourcentage d’erreur par rapport aux valeurs de référence correspondantes étant respectivement de 11 % et 13 %. Cela signifie que le nombre d’échantillons doit être augmenté pour obtenir un meilleur niveau de précision. Néanmoins, cette observation peu suffire pour identifier les paramètres de plus forte influence. Cette conclusion est également étayée par la dispersion des indices totaux de ces variables, qui est plus élevée que celle liée au reste des variables d’entrée. Le nombre de variables d’entrée étant inférieur à 10, il est néanmoins possible de tirer des conclusions.

En ce qui concerne les indices de Shapley, leurs valeurs sont comprises entre S_i et S_{T_i} , et leur somme est égale à un. Globalement, ils fournissent la même information que S_{T_i} . En ce qui concerne la fiabilité des indices calculés, l’incertitude des indices de Shapley est plus élevée que celle des indices de Sobol, en accord avec les résultats obtenus pour la fonction d’Ishigami.

Conclusion intermédiaire 2 *Ce second cas test est intéressant par la nature de son*

observable. En effet, dans le premier problème il s'agit d'observer une sortie classique, à savoir une position x à un instant t . Dans ce second cas test, la sortie est booléenne, pour qualifier si le comportement du système est stable ou pas. De plus, ce modèle de *Stick-Slip* présente une forte non-linéarité dans l'expression de sa force de frottement. Malgré ces 2 particularités, l'ASG a pu être menée à bien et les résultats fournis sont cohérents. Cela démontre le large domaine d'utilisation de ce type de méthode.

2.5 Analyse de sensibilité globale sur des modèles EF

2.5.1 Problème 1 : Analyse statique d'une structure treillis

2.5.1.1 Formulation du problème et modèle numérique

Dans les cas tests examinés dans les sections 2.3 et 2.4, l'ASG est appliquée à un benchmark analytique et aux études de systèmes mécaniques via une approche MBSE. Dans cette section, l'ASG est réalisée sur un modèle de structure en treillis dont la réponse mécanique est obtenue via un modèle EF dédié. L'objectif est d'évaluer l'influence des aires des sections transversales des poutres sur le déplacement vertical de la structure, au nœud où la force est appliquée.

La géométrie de la structure en treillis, représentée sur la Fig. 2.10, est tirée de [189]. Les propriétés géométriques et du matériau, utilisées dans les analyses numériques, sont énumérées dans le Tab. 2.4. En particulier, les propriétés des matériaux, module d'Young E et coefficient de Poisson ν , sont maintenues constantes ainsi que les longueurs des poutres constituant la structure, L_i . Une force verticale F est appliquée au nœud 7 de la Fig. 2.10 et les degrés de liberté sont fixés à 0 aux nœuds 1 et 4. Pour reproduire exactement la formulation du problème présentée dans [189], les aires des sections transversales, considérées comme variables d'entrée pour l'ASG, sont celles des poutres 1 à 9. Le Tab. 2.4 indique l'intervalle de variation des sections transversales considérées comme variables ainsi que la valeur de la section transversale de la poutre 10, qui est constante. Le modèle paramétrique EF de la structure en treillis a été généré avec le langage de conception paramétrique APDL dans le logiciel commercial ANSYS® EF. Chaque poutre est modélisée avec des éléments LINK180 (éléments de treillis avec trois degrés de liberté par nœud). Enfin, des analyses statiques linéaires par EF ont été effectuées.

2.5.1.2 Résultats numériques

Les résultats de l'analyse de Sobol, compte tenu de la méthode FAST, sont présentés dans la Fig. 2.11 et comparés aux valeurs de référence des indices élémentaires S_i tirées de [189]. En ce qui concerne les résultats des indices totaux, les valeurs de référence correspondantes sont obtenues en utilisant l'algorithme de Saltelli 2010 avec $N = 2^{22}$. On peut déduire de ces résultats que l'influence croisée entre les variables est négligeable car

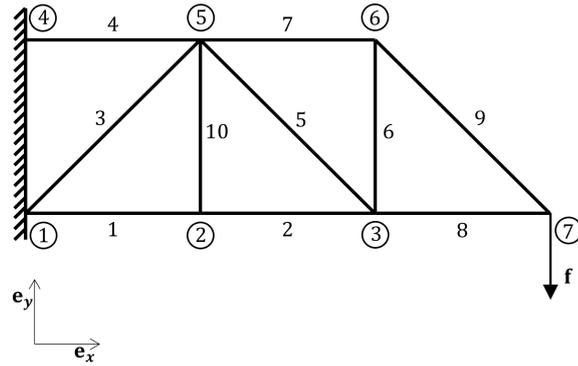


FIGURE 2.10 – Modèle de structure en treillis.

TABLE 2.4 – Variables constantes et variables d'entrée : valeurs de référence et intervalles de variation.

Variable	Référence	Intervalle
E [MPa]	2.1×10^5	constant
ν [-]	0.3	constant
$L_{i \in \{1,2,4,7,8,10\}}$ [mm]	1×10^3	constant
$L_{i \in \{3,5,6,9\}}$ [mm]	$\sqrt{2} \times 10^3$	constant
$A_{\text{sec}_{10}}$ [mm ²]	1×10^2	constant
$A_{\text{sec}_{i \in \{1, \dots, 9\}}}$ [mm ²]	1×10^2	$\mathcal{U}([0.5, 2.5]) \times 10^2$
F [N]	1000	constant

les valeurs des indices élémentaires et totaux sont très proches. En effet, la somme des influences corrélées est d'environ 2 %.

De plus, les résultats de la Fig. 2.11 soulignent que la variable d'entrée ayant la plus forte influence sur le déplacement vertical maximal est la section transversale de la poutre 4, A_{sec_4} , avec des valeurs de S_i et S_{T_i} respectivement égales à 56 % et 58 %. Inversement, les sections transversales ayant les plus faibles influences sont celles des poutres 6, 7 et 8 avec des valeurs de S_i et S_{T_i} inférieures à 1.1 %.

En comparant l'erreur relative entre les valeurs des indices calculés et celles utilisées comme références, on constate qu'un nombre d'échantillons $N = 2^{12}$ est suffisant pour limiter l'erreur moyenne en pourcentage à un maximum de 2.5 %. En ce qui concerne les indices de Shapley, les mêmes remarques que celles qui ont été formulées pour les cas analytiques présentés dans les sections précédentes peuvent être répétées ici.

Conclusion intermédiaire 3 *Ce modèle EF d'une structure en treillis est intéressant pour son nombre important de variables, i.e., 9 contre 5 dans les cas précédents. Son nombre de variables d'étude est intéressant d'autant plus qu'il correspond à la limite de fonctionnement de l'algorithme FAST, à savoir 10, déduit par Tissot et Prieur [177]. En définitif, les résultats restent cohérents et interprétables bien que l'écart avec la solution de référence ne soit pas négligeable.*

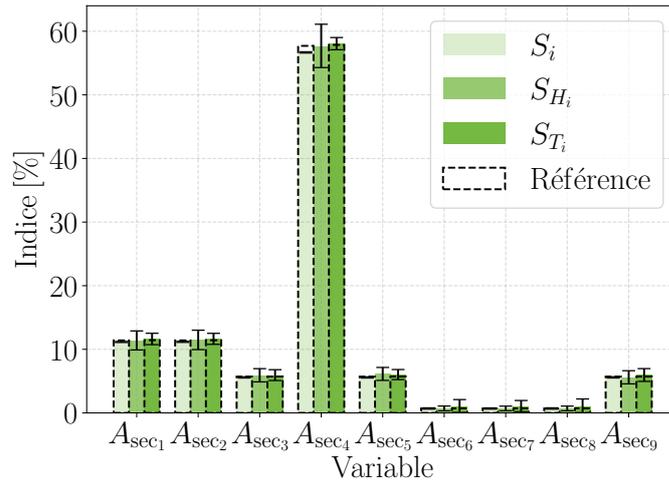


FIGURE 2.11 – Résultats de l'Analyse de Sensibilité Globale du modèle treillis.

2.5.2 Problème 2 : Analyse de flambement par valeurs propres d'un panneau raidi en composite mono-raisseur.

2.5.2.1 Formulation du problème et modèle numérique

L'utilité de l'ASG est démontrée à travers l'étude d'un problème d'ingénierie réaliste portant sur un panneau raidi composite soumis à une compression uni-axiale, une solution structurelle généralement adoptée dans les structures aéronautiques.

La géométrie du panneau raidi utilisé pour effectuer l'ASG est tirée de [129] et est illustrée en Fig. 2.12 avec les paramètres géométriques pertinents utilisés dans l'analyse. La recherche de compromis entre une modélisation simplifiée et un comportement mécanique adéquat (en particulier lorsque le flambement est le principal phénomène critique impliqué dans la formulation du problème) n'est pas une tâche triviale si la géométrie du raidisseur et les séquences d'empilement de celui-ci et du panneau sont utilisées comme variables de conception. Même si des règles de conception ont été définies pour limiter la complexité liée à la conception de ces structures, le grand nombre de variables de conception (géométrie et matériaux) conduit à un nombre important de solutions structurelles à évaluer. En outre, sans les limites des règles de conception standard, il est possible d'obtenir une amélioration remarquable de l'efficacité structurelle en abordant le problème via une approche d'optimisation multi-échelle, comme décrit dans [129]. Néanmoins, l'ASG peut être utilisée comme outil de conception préliminaire pour déterminer s'il est possible d'évaluer la sensibilité des réponses structurelles à l'ensemble des variables de conception. Ainsi, il sera possible de déterminer les paramètres qui peuvent être exclus de l'ensemble des variables de conception en raison de leur influence négligeable sur le comportement de la structure. Par conséquent, la réalisation d'une ASG préliminaire permettra de réduire l'effort de calcul des campagnes numériques ultérieures (y compris le processus d'optimisation).

Dans ce contexte, l'objectif de l'exemple présenté dans cette section est d'évaluer l'influence des variables géométriques, illustrées dans la Fig. 2.12, et des paramètres polaires décrivant le comportement anisotrope des matrices de rigidité du stratifié à l'échelle macroscopique [124, 129] sur la valeur de la première charge de flambement du panneau raidi.

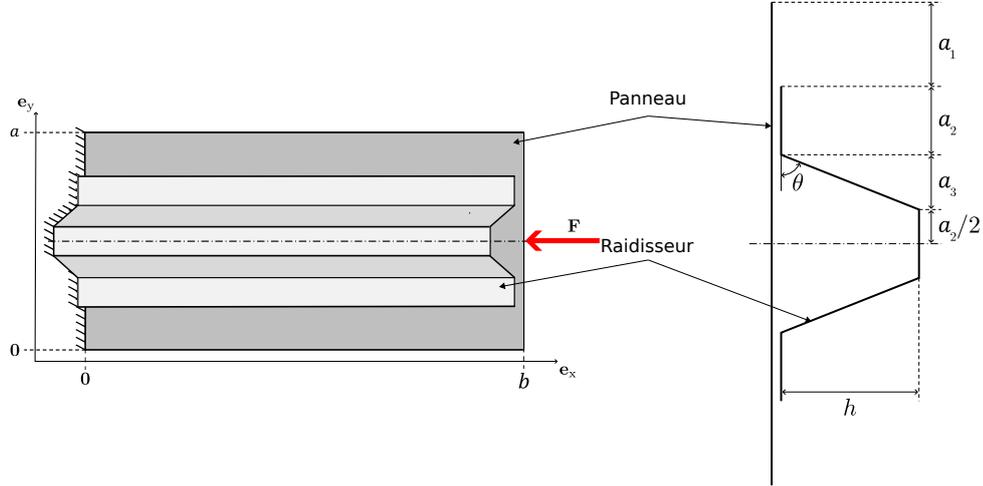


FIGURE 2.12 – Schéma d'un panneau composite raidi.

Le panneau raidi est constitué de deux régions, le panneau et le raidisseur. Le panneau, caractérisé par une longueur a et une largeur $b = 2(a_1 + a_2 + a_3)$, est constitué de n_S plis pré-impregnés carbone-époxy, dont les propriétés matérielles sont données dans [129]. Le raidisseur dont la géométrie est entièrement définie par les paramètres géométriques a_2 , a_3 et h , est constitué de n_B plis. Des quantités normalisées sont introduites pour définir la géométrie du panneau raidi comme suit :

$$c_1 = 2\frac{a_2}{a}, \quad c_2 = 2\frac{a_3}{a_2}, \quad c_3 = \frac{h}{a_2}. \quad (2.25)$$

Pour éviter de générer des géométries incohérentes, deux inégalités sont introduites comme suit :

$$a_1 = \frac{a}{2} - a_2 - a_3, \quad \theta = \text{atan}\left(\frac{h}{a_3 - \frac{a_2}{2}}\right), \quad \text{avec } a_1 > 0, \quad (2.26)$$

$$a_3 \geq \frac{a_2}{2},$$

La première inégalité est nécessaire pour éviter le chevauchement entre les raidisseurs voisins et la seconde empêche les valeurs négatives de l'angle θ . Elles permettent également de définir les intervalles de variations reportés dans le Tab. 2.5. Le comportement mécanique des stratifiés est défini par le formalisme polaire, qui est une approche efficace pour définir la réponse mécanique des stratifiés composites [124], en particulier à des fins de conception. Plus précisément, comme indiqué dans [124], seuls trois paramètres polaires sont nécessaires pour décrire le comportement élastique macroscopique (en termes de ma-

trices de rigidité de membrane, de flexion et de couplage membrane-flexion) d'un stratifié orthotrope quasi-homogène : deux modules anisotropes, à savoir, R_{0K}^{A*} et R_1^{A*} , et l'angle polaire Φ_1^{A*} (qui représente l'orientation de l'axe principal d'orthotropie). Bien entendu, la valeur maximale de la première charge de flambement peut être obtenue en alignant l'axe principal d'orthotropie avec la direction de la charge appliquée. Pour satisfaire cette condition, il suffit de fixer $\Phi_1^{A*} = 0$ pour le raidisseur et le panneau, comme indiqué dans [129] ; ainsi, Φ_1^{A*} n'est pas intégré parmi les variables de conception du problème. Comme pour les paramètres géométriques, des valeurs normalisées sont également utilisées pour définir le comportement mécanique du panneau raidi. En particulier, deux coefficients sont introduits comme suit :

$$\rho_0 = \frac{R_{0K}^{A*}}{R_0}, \quad \rho_1 = \frac{R_1^{A*}}{R_1}, \quad (2.27)$$

où R_0 et R_1 sont les modules anisotropes de la matrice de rigidité réduite du pli [129]. En outre, comme le suggère [93], il est possible d'éviter d'introduire les conditions de faisabilité sur ρ_0 et ρ_1 (qui sont nécessaires pour garantir l'existence d'un empilement correspondant aux valeurs des modules anisotropes et des angles polaires utilisés à l'échelle macroscopique pour décrire le comportement anisotrope du stratifié) en envisageant le changement de variable suivant :

$$(\alpha_0, \alpha_1) := \left(\frac{\rho_0 - 1}{2(\rho_0^2 - 1)}, \rho_1 \right). \quad (2.28)$$

Bien entendu, différentes valeurs de α_0 et α_1 sont utilisées pour le panneau et pour le raidisseur, à savoir α_{0_S} et α_{1_S} pour le panneau et α_{0_B} et α_{1_B} pour le raidisseur. Pour plus de détails sur la mise en œuvre du formalisme polaire dans l'analyse par éléments finis des structures composites, voir [93, 129].

La configuration de référence du panneau raidi mono-raisseur est caractérisée par une valeur de la première charge de flambement égale à 445074 N [129] obtenue avec les valeurs de référence du Tab. 2.5. Ce tableau contient également les variables d'entrée de l'ASG ainsi que leur plage de variation. Une description détaillée de la stratégie numérique visant à récupérer les séquences d'empilement à partir de la définition des paramètres polaires du stratifié est proposée dans [129].

Le modèle EF du panneau raidi, représenté dans la Fig. 2.13, est créé via des scripts APDL et son maillage est constitué d'éléments SHELL181 (éléments à quatre nœuds avec six degrés de liberté à chaque nœud). Un soin particulier a été apporté à la définition de la taille du maillage afin d'obtenir au moins trois éléments le long de la largeur de chaque panneau horizontal et de la partition des raidisseurs (segments de longueur a_1 , a_2 et a_3) et au moins six éléments pour les segments de hauteur h . Pour le panneau raidi de référence, le modèle EF consiste en un total de 4438 éléments. Pour imposer des conditions limites périodiques entre les bords opposés du panneau, des contraintes appropriées ont été utilisées (voir [129] pour plus de détails sur ce point). En outre, des éléments de contrainte multi-points rigides (MPC184) ont été créés et reliés à deux nœuds de référence des deux

côtés : à $x = 0$, le nœud de référence concerné est encastré tandis que, à $x = b$, tous les degrés de liberté ont été mis à zéro, à l'exception du déplacement le long de l'axe x . En outre, une force unitaire le long de l'axe x est appliquée sur ce nœud. Tous les détails concernant le modèle EF du panneau raidi sont disponibles dans [129].

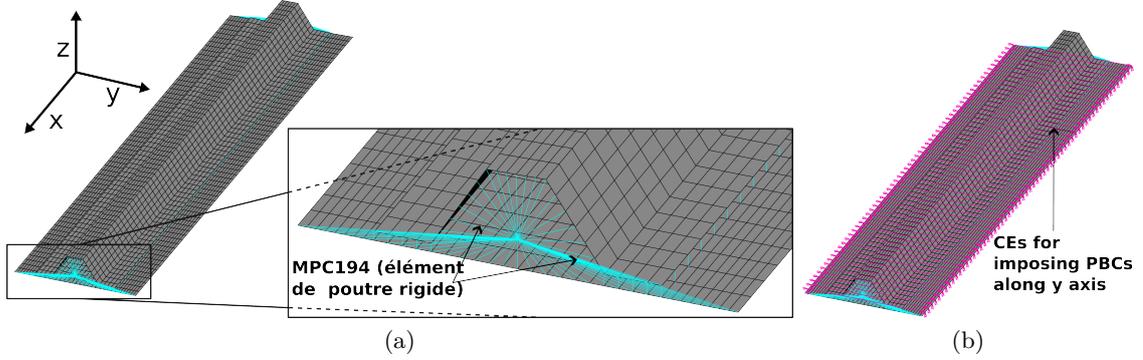


FIGURE 2.13 – (a) Modèle aux Éléments Finis de l'unité répétée du panneau raidi avec détails des éléments MPC184 [?] et (b) les contraintes pour imposer les conditions limites périodiques le long de l'axe y .

2.5.2.2 Résultats numériques

Les résultats de l'analyse de Sobol obtenus avec l'algorithme FAST pour $N = 2^{12}$ sont présentés dans la Fig. 2.14 et comparés aux résultats de référence obtenus avec la méthode Saltelli 2010 pour $N = 2^{22}$. En observant ces résultats, on peut déduire que l'influence des interactions entre les variables d'entrée est importante avec une valeur approximativement égale à 24 %.

TABLE 2.5 – Variables constantes et variables d'entrée : valeurs de référence et intervalles de variation.

Variable	Référence	Intervalle	Type
n_S [-]	28	$\mathcal{U}([20, 32])$	Discrete
n_B [-]	32	$\mathcal{U}([20, 32])$	Discrete
c_1 [-]	0.2	$\mathcal{U}([0.1, 0.45])$	Continue
c_2 [-]	2.87	$\mathcal{U}([1, 3])$	Continue
c_3 [-]	2	$\mathcal{U}([1, 3])$	Continue
α_{0S} [-]	0.079	$\mathcal{U}([0, 1])$	Continue
α_{1S} [-]	0.57	$\mathcal{U}([0, 1])$	Continue
α_{0B} [-]	0.044	$\mathcal{U}([0, 1])$	Continue
α_{1B} [-]	0.50	$\mathcal{U}([0, 1])$	Continue

Les résultats de la Fig. 2.14 montrent que c_1 est la variable dont l'indice total S_{T_i} est le plus élevé (approximativement égal à 61 %). Dans les faits, c_1 agit sur la largeur des *flanges* du panneau, ce qui a un impact important sur la rigidité en flexion du panneau. De plus, cette même variable est aussi celle dont l'interaction avec les autres variables est la plus forte puisque la différence entre les contributions élémentaires et totales des

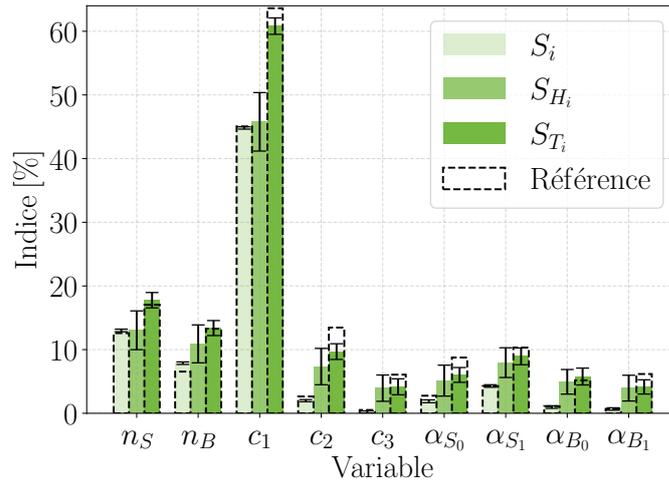


FIGURE 2.14 – Résultats de l’Analyse de Sensibilité Globale sur le modèle du panneau raidi.

indices de Sobol pour c_1 est de l’ordre de 15 %. Inversement, les variables pour lesquelles l’ASG retourne les valeurs les plus faibles des indices totaux et élémentaires sont c_3 et les quantités α_{iS} , α_{iB} ($i = 0, 1$) avec des valeurs inférieures à 10 %. Il convient de noter que la réponse au flambement peut être très sensible aux paramètres géométriques et polaires. C’est pourquoi, en lisant les résultats de la Fig. 2.14, il faut se rappeler que, bien que la quantité de sortie utilisée dans le cadre de l’ASG soit la valeur propre, différents ensembles de paramètres d’entrée peuvent produire différentes formes de mode de flambement. À titre d’exemple, les formes de mode de flambement obtenues avec quatre ensembles de variables d’entrée sont présentées dans la Fig. 2.15. En particulier, les valeurs extrêmes des intervalles de variation des variables d’entrée c_1 et c_3 ont été prises en compte, tandis que les autres paramètres sont égaux à ceux de la configuration de référence, dont les détails sont disponibles dans [129].

Les résultats de la Fig. 2.14 montrent également que pour les variables c_2 , c_3 et α_{iS} , α_{iB} ($i = 0, 1$), le pourcentage d’erreur des indices de Sobol par rapport aux valeurs de référence se situe entre 28 % et 32 %. Ce résultat suggère qu’un plus grand nombre d’échantillon devrait être utilisée bien que l’identification des paramètres influents puissent être menée. En ce qui concerne les valeurs des indices de Shapley de la Fig. 2.14, les mêmes remarques que celles faites pour les autres cas de test s’appliquent également à ce cas.

Conclusion intermédiaire 4 *Ce modèle EF de panneau raidi mono-raidisseur est un cas d’étude initialement employé comme cas d’application d’une méthode d’optimisation multi-échelle [129]. L’intérêt du modèle est lié au nombre de paramètres, 9, qui est, comme précédemment, proche de la limite de fonctionnement de l’algorithme FAST. Néanmoins, les résultats tirés de cette analyse sont cohérents et exploitables bien qu’ils comportent un écart non négligeable avec les résultats de référence. Par ailleurs, ce cas test montre qu’une ASG peut être utilisée pour définir les variables d’étude les plus influents à prendre*

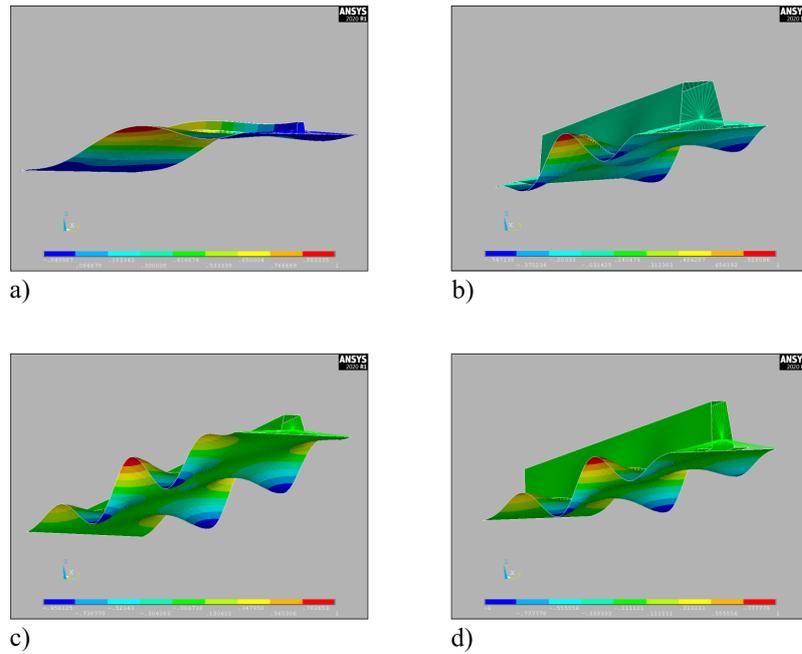


FIGURE 2.15 – Formes de mode de flambement du panneau raidi à un seul raidisseur et en considérant les valeurs de référence pour les paramètres avec en (a) $c_1 = c_{1\min}$, (b) $c_1 = c_{1\max}$, (c) $c_3 = c_{3\min}$ et (d) $c_3 = c_{3\max}$.

en compte dans l'optimisation multi-échelle et ceux à négliger.

2.6 Conclusion du chapitre

Dans ce travail, l'efficacité de différents algorithmes pour l'ASG a été évaluée sur des problèmes à entrées multiples de nature différente. Un benchmark analytique a été utilisé pour comparer les différents algorithmes disponibles dans la littérature, basés à la fois sur les indices de Sobol et l'effet Shapley. Cette comparaison a été réalisée par rapport aux coûts de calcul, aux précisions et aux incertitudes. En ce qui concerne les indices de Sobol, un compromis adéquat entre les critères considérés a été obtenu avec l'algorithme FAST. En outre, les indices de Shapley ont également été utilisés car ils fournissent des informations complémentaires par rapport aux indices de Sobol. L'algorithme développé par Goda et traduit en Python a été utilisé. L'ASG a ensuite été appliquée sur deux problèmes impliquant des systèmes dynamiques fortement non linéaires temporellement, à savoir deux modèles d'oscillateurs modélisés via le langage Modelica. Le premier est un oscillateur classique dont on étudie le comportement durant sa phase de transition vers sa solution d'équilibre, i.e., sa position pour un temps donné t . Le second est un oscillateur à frottement sec dont la force de frottement est fortement non-linéaire, voire discontinue selon son expression. Le but est d'étudier sa stabilité via une sortie booléenne. Pour réduire davantage le coût de calcul requis par les algorithmes considérés, les modèles Modelica ont été exportés via la norme FMI puis couplés aux algorithmes d'ASG codés

dans l'environnement Python.

Des ASG ont également été menées sur deux problèmes structurels caractérisés par un plus grand nombre de variables par rapport aux autres problèmes étudiés. Ce nombre de variables étant de 9, cela a permis d'observer le comportement de l'algorithme FAST à la limite de son environnement d'utilisation. Ces problèmes ont été résolus via des analyses aux EF. Le premier modèle correspond à l'analyse linéaire statique d'une structure en treillis. Le second cas test correspond à l'analyse de flambement aux valeurs propres d'un panneau raidi en matériau composite. Ces deux modèles numériques ont été créés via des scripts APDL paramétriques et couplés aux algorithmes basés sur les indices de Sobol et l'effet Shapley.

Les résultats de l'ASG sur les problèmes mécaniques ont mis en évidence, d'une part, les variables d'entrée ayant la plus faible influence sur la quantité observée et, d'autre part, l'existence d'interactions entre les variables considérées. En fonction de l'objectif défini par l'utilisateur, ces résultats préliminaires peuvent être utilisés pour différentes applications :

- réduire le nombre de variables à prendre en compte dans l'étude du modèle. Cela peut être utile dans une optique d'optimisation. Si des variables de conception se révèlent peu influentes, il n'est pas nécessaire de les inclure dans la stratégie d'optimisation ;
- identifier les variables de conception d'un modèle pour accroître leurs précisions de dimensionnement.

Dans le cadre de cette thèse, l'objectif va être d'identifier les paramètres influents mais surtout de déduire les sous-systèmes influents du modèle global MBSE. Pour y parvenir, une simple ASG n'est pas suffisante dans la mesure où elle fournit des informations sur l'influence des variables de conception et non pas sur les influences des sous-systèmes. Ainsi, cet outil mathématique doit être couplé à un avis critique d'ingénieur pour aider à la décision.

Ce chapitre représente la première étape dans le développement de la méthodologie complète de modélisation multifidélité incluant des métamodèles basés sur les hypersurfaces NURBS. Une fois les sous-systèmes critiques et influents identifiés, le but va être de les enrichir pour accroître la précision du modèle MBSE utilisé pour rassembler les réponses des différents sous-systèmes constituant le modèle global. Cet enrichissement va se faire par l'inclusion d'un métamodèle basé sur une nouvelle stratégie faisant intervenir des entités géométriques, les NURBS, pour émuler des systèmes multiphysiques complexes caractérisés par plusieurs centaines de variables de conception (lors de l'optimisation des valeurs des poids des *Control Points* par exemple). Grâce à une utilisation intelligente des modèles aux EF de haute fidélité, voire de données expérimentales, pour construire et valider le métamodèle, il sera possible de simuler avec précision des phénomènes complexes tout en réduisant considérablement les coûts de calcul. Ces modèles devront fonctionner en

temps quasi-réel, de la même manière que les autres sous-systèmes Modelica. Le chapitre suivant présente cette stratégie de métamodélisation comme la solution d'un problème d'optimisation sous contraintes avant de l'appliquer à des problèmes pertinents.

3

Développement d'un métamodèle basé sur les entités Non-Uniform Rational Basis-Spline

Sommaire

3.1	Introduction	85
3.2	Généralités sur les hypersurfaces NURBS	86
3.3	Génération du métamodèle basé sur les entités NURBS comme solution d'un problème d'optimisation	88
3.3.1	Description du problème	88
3.3.2	Initialisation du problème	89
3.3.3	Énoncé du problème d'optimisation	91
3.4	Mise en place du processus d'optimisation	95
3.4.1	Résolution du <i>Constrained Non-Linear Programming Problem</i> (CNLPP) pour l'évaluation des coordonnées des CPs	95
3.4.1.1	Formulation générale	95
3.4.1.2	Formulation du problème d'approximation d'hypersurface par séparation de variables	97
3.4.1.3	Choix de la formulation	99
3.4.2	Optimisation numérique	101
3.4.2.1	Optimisation des vecteurs nodaux	103
3.4.2.2	Optimisation des poids	108
3.5	Application du métamodèle sur des cas analytiques représentatifs	110
3.5.1	Influence du lissage sur les points de contrôle	111
3.5.2	Influence de l'optimisation des vecteurs nodaux et <i>overfitting</i>	113
3.5.2.1	Optimisation des vecteurs nodaux	113
3.5.2.2	Réduction du phénomène d' <i>overfitting</i>	115
3.5.3	Influence de l'optimisation du poids	115
3.6	Conclusion	117

RÉSUMÉ DU CHAPITRE

Initialement employées dans la Conception Assistée par Ordinateur (CAO), les entités Non-Uniform Rational Basis-Spline (NURBS) sont apparues comme un outil prometteur dans la métamodélisation. Leur grande maniabilité en fait des métamodèles capables de s'adapter à une grande variété de problèmes Multiple-Input-Multiple-Output (MIMO), linéaires ou non. En contrepartie, cette souplesse a un défaut : le paramétrage de l'entité NURBS n'est pas une tâche anodine et obtenir un métamodèle précis et fiable peut se révéler être un vrai défi. L'enjeu de cette thèse est de proposer une méthode d'optimisation systématique pour pouvoir calibrer au mieux les valeurs des différents paramètres régissant l'allure de l'hypersurface NURBS. Ainsi, la génération d'un métamodèle basé sur une hypersurface NURBS se pose comme un problème d'optimisation sous contraintes, avec prise en compte, si besoin, d'un terme de lissage. Ce terme a pour but de réduire le phénomène d'overfitting et de minimiser les éventuelles oscillations locales de la base de données, assimilables à du bruit, pouvant provenir d'instabilités inhérentes au modèle initial, à des incertitudes de mesure ou encore à un manque de précision numérique.

3.1 Introduction

La stratégie de métamodélisation présentée dans ce travail repose sur l'utilisation d'hypersurfaces NURBS dans le domaine de définition à N dimensions (dimension de l'espace des entrées) et le co-domaine à M dimensions (dimension de l'espace des sorties) pour approximer un ensemble de points de données appelés TPs. Initialement employées comme simples entités géométriques dans les logiciels de CAO dans le domaine de l'optimisation de forme [52, 138], l'utilisation des entités NURBS pour formuler un modèle réduit, ou métamodèle, est assez récente [10, 11, 178]. Une hypersurface NURBS est une entité polynomiale qui généralise les hypersurfaces B-splines, elles-mêmes généralisant les hypersurfaces de Bézier [138]. Cette généralisation permet aux entités NURBS d'être pilotées par cinq types de paramètres [138] comprenant : le degré des polynômes de Bernstein, le nombre de CPs, les coordonnées des CPs, les poids associés aux CPs et les composantes non-triviales des vecteurs nodaux le long de chaque dimension d'entrée. Ces paramètres peuvent être utilisés pour contrôler la forme, le lissage ou encore les non-linéarités potentielles de l'hypersurface [11, 18].

La formulation présentée dans ce travail vise à généraliser la stratégie de métamodélisation, basée sur les hypersurfaces NURBS, proposée par Audoux et al [11]. Dans [11], un algorithme d'optimisation hybride est proposé pour optimiser tant les paramètres entiers que continus, impliqués dans la définition des hypersurfaces NURBS. Cette méthodologie hybride utilise un Algorithme Génétique (AG) combiné à un algorithme au gradient pour résoudre un problème des moindres carrés. Le premier est employé pour optimiser les variables entières que sont le degré des polynômes de Bernstein et le nombre total de CPs. L'algorithme au gradient est utilisé pour déterminer les composantes non-triviales des vecteurs nodaux et les valeurs des poids associés aux CPs. Pendant les deux phases d'optimisation, les coordonnées des CPs sont calculées avec une formule analytique qui nécessite un processus numérique complexe pour être implémentée dans le cas des hypersurfaces ($N > 2$). Cependant, dans [11], le calcul du gradient de la fonction objectif à minimiser se base sur un calcul par différences finies : les coûts de calculs sont importants et l'algorithme utilisé pour calculer la valeur optimale des coordonnées des CPs (selon la formule analytique) manque de généralité et d'efficacité.

Ce travail propose une généralisation aux cas MIMO des algorithmes employés pour la résolution des problèmes de minimisation aux moindres carrés de [9]. Un terme de lissage, exprimé par [18, 56] pour le cas des problèmes de *surface fitting*, et généralisé au cas des hypersurfaces dans cette thèse, est également introduit dans la résolution analytique du problème aux moindres carrés intervenant dans la génération des coordonnées optimales des CPs. En outre, le calcul du gradient utilisé dans l'algorithme au gradient, est reformulé analytiquement, pour s'affranchir des contraintes introduites par la méthode des différences finies. Ces gradients analytiques sont tirés de [18] et généralisés aux cas des hypersurfaces.

Pour finir, un autre des apports de cette thèse est la formulation du métamodèle basé sur les entités NURBS via une séparation des dimensions, introduite initialement par [11] dont l'objectif est de réduire les coûts de calculs nécessaires à la génération d'un tel métamodèle.

Pour présenter la génération du métamodèle basé sur les hypersurfaces NURBS, ce chapitre s'organise comme suit. La formulation d'une hypersurface NURBS classique et son emploi dans une stratégie de métamodélisation sont rappelés et définis dans la section 3.2. La formulation de la génération du métamodèle en tant que solution d'un problème d'optimisation est donnée dans la section 3.3. La stratégie d'optimisation mise en place pour générer le métamodèle est présentée dans la section 3.4. Le détail et le rôle de chaque étape du processus d'optimisation sont discutés dans la section 3.5. Pour finir, la section 3.6 conclue ce chapitre et propose différentes perspectives.

3.2 Généralités sur les hypersurfaces NURBS

Cette section rappelle la formulation des entités NURBS. Leur formulation sous forme de courbes et surfaces est présentée dans [18, 39, 138] et peut facilement être déduite de la formulation générale exposée ci-après. Utilisées à l'origine par les logiciels CAO dans les années 1990 en tant qu'outils de modélisation de courbes et surfaces, ces entités sont maintenant utilisées dans plusieurs domaines tels que l'optimisation topologique [38, 149], l'optimisation de forme [17, 128], l'optimisation des distributions des invariants élastiques régissant l'anisotropie des composites à rigidité variable [28, 53, 93, 94, 127] ou encore leur utilisation en tant que fonctions de base dans des méthodes EF [84]. Leur emploi pour la génération de métamodèles est en pleine émergence [10, 11] et fait l'objet de ce travail.

Pour un problème MIMO, une hypersurface NURBS est une fonction vectorielle définie sur un domaine paramétrique de dimension N , vers un co-domaine de dimension M par l'application : $\mathbf{h} : \mathbb{R}^N \mapsto \mathbb{R}^M$ [11]. L'objectif du métamodèle est d'approcher une base de données \mathbf{Q} de dimension $N + 1$, stockée sous la forme d'un *array* de taille $M \times (r_1 + 1) \times \dots \times (r_N + 1)$, où $(r_k + 1)$ est la longueur de la matrice dans la direction d'entrée $k = 1, \dots, N$. Le terme $(r_k + 1)$ correspond également à la longueur du vecteur \mathbf{u}_k contenant les coordonnées du paramètre k du modèle ayant généré la base de données \mathbf{Q} . En d'autres termes, la matrice \mathbf{Q} contient $n_{\text{TP}} = \prod_{k=1}^N (r_k + 1)$ TPs et chaque TP a M coordonnées. Les TPs proviennent d'expériences ou de résultats de simulations numériques haute-fidélité pilotées par N paramètres. Il est primordial de relever que dans ce travail, les jeux de paramètres servant à générer les bases de données sont répartis dans l'espace selon une grille uniforme. En considérant les notations introduites dans [11, 138],

l'expression générale d'une hypersurface NURBS est donnée par :

$$\mathbf{h}(\zeta_1, \dots, \zeta_N) = \sum_{i_1=0}^{n_1} \cdots \sum_{i_N=0}^{n_N} R_{i_1, \dots, i_N}(\zeta_1, \dots, \zeta_N) \mathbf{P}_{i_1, \dots, i_N}, \quad (3.1)$$

avec le vecteur $\mathbf{h} \in \mathbb{R}^M$ correspondant aux coordonnées de l'hypersurface NURBS dans l'espace des sorties, évaluée au jeu de coordonnées paramétriques normalisées $(\zeta_1, \dots, \zeta_N)$ et l'*array* \mathbf{P} contenant les CPs, chaque CP étant défini par ses coordonnées dans l'espace des sorties selon un vecteur de taille \mathbb{R}^M . L'*array* \mathbf{P} est de taille $M \times (n_1 + 1) \times \cdots \times (n_N + 1)$, où $(n_k + 1)$ est la taille de l'*array* dans la direction d'entrée $k = 1, \dots, N$. En d'autres termes, l'*array* \mathbf{P} contient $n_{\text{CP}} = \prod_{k=1}^N (n_k + 1)$ CPs, chaque CP ayant M coordonnées. Quant à $R_{i_1, \dots, i_N}(\zeta_1, \dots, \zeta_N)$, il s'agit de la fonction rationnelle de base liée au CP générique exprimée par :

$$R_{i_1, \dots, i_N}(\zeta_1, \dots, \zeta_N) = \frac{N_{i_1, s_1}(\zeta_1) \times \cdots \times N_{i_N, s_N}(\zeta_N) \omega_{i_1, \dots, i_N}}{\sum_{j_1=0}^{n_1} \cdots \sum_{j_N=0}^{n_N} (N_{j_1, s_1}(\zeta_1) \times \cdots \times N_{j_N, s_N}(\zeta_N) \times \omega_{j_1, \dots, j_N})}, \quad (3.2)$$

avec ω l'*array* de taille n_{CP} , contenant les poids $\omega_{i_1, \dots, i_N} > 0$, respectivement associés à chaque CP. Enfin, N_{i_k, s_k} sont les polynômes de Bernstein de degré s_k , évalués récursivement avec l'algorithme de Cox et De Boor [138] défini par :

$$\begin{cases} N_{i_k, 0}(\zeta_k) = \begin{cases} 1, & \text{si } v_{i_k} < \zeta_k < v_{i_k+1}, \\ 0, & \text{sinon,} \end{cases} \\ N_{i_k, q_k}(\zeta_k) = \frac{\zeta_k - v_{i_k}}{v_{i_k+q_k} - v_{i_k}} N_{i_k, q_k-1}(\zeta_k) + \frac{v_{i_k+q_k+1} - \zeta_k}{v_{i_k+q_k+1} - v_{i_k+1}} N_{i_k+1, q_k-1}(\zeta_k), \end{cases} \quad (3.3)$$

avec $q_k = 1, \dots, s_k$, ζ_k la coordonnée paramétrique normalisée entre 0 et 1 dans la direction d'entrée k et v_{i_k} la i_k -ième composante (appelée aussi nœud) du vecteur-nodal monotone non-décroissant de longueur $m_k + 1$:

$$\mathbf{v}_k^T = \underbrace{\{0, \dots, 0\}}_{s_k+1}, v_{p_k+1}, \dots, v_{m_k-s_k-1}, \underbrace{\{1, \dots, 1\}}_{s_k+1}, \mathbf{v}_k \in \mathbb{R}^{m_k} \text{ avec } m_k = n_k + s_k + 1. \quad (3.4)$$

Dans la suite de ce document, les nœuds $v_{i_k} \neq 0, 1$ sont appelés nœuds non triviaux, ou nœuds internes, par opposition aux nœuds triviaux $v_{i_k} = 0, 1$ ou nœuds externes. Le vecteur nodal permet de définir le domaine de variation des fonctions de base, i.e., des polynômes de Bernstein, non nuls sur un intervalle de $s_k + 1$ nœuds successifs. En outre, les nœuds triviaux contraignent les entités NURBS à interpoler exactement les TPs qui bornent le domaine de variation de l'espace des sorties de dimensions \mathbb{R}^M [138]. Les polynômes de Bernstein définis par l'Eq. (3.3), possèdent différentes propriétés définies dans [138], dont la règle de partition de l'unité :

$$\sum_{i_k}^{n_k} N_{i_k, s_k} = 1. \quad (3.5)$$

En considérant cette propriété, lorsque les poids de l'hypersurface NURBS prennent la même valeur, l'expression de l'hypersurface NURBS de l'Eq. (3.1) se simplifie par :

$$\mathbf{h}(\zeta_1, \dots, \zeta_N) = \sum_{i_1=0}^{n_1} \dots \sum_{i_N=0}^{n_N} N_{i_1, s_1}(\zeta_1) \dots N_{i_N, s_N}(\zeta_N) \mathbf{P}_{i_1, \dots, i_N}, \quad (3.6)$$

ce qui correspond à une hypersurface B-spline. Une autre propriété intéressante des polynômes de Bernstein est leur propriété de support local. Cela permet à un point de l'hypersurface NURBS d'être dépendant seulement des CPs qui l'entourent. Formellement, cela se traduit par :

$$R_{i_1, \dots, i_N}(\zeta_1, \dots, \zeta_N) \neq 0 \text{ si } (\zeta_1, \dots, \zeta_N) \in [v_{i_1}^{(1)}, v_{i_1+s_1+1}^{(1)}[\times \dots \times [v_{i_N}^{(N)}, v_{i_N+s_N+1}^{(N)}[, \quad (3.7)$$

où l'hyper-rectangle ouvert $[v_{i_1}^{(1)}, v_{i_1+s_1+1}^{(1)}[\times \dots \times [v_{i_N}^{(N)}, v_{i_N+s_N+1}^{(N)}[$ est la portion du domaine global \mathbb{R}^N affectée par le CP de coordonnées $\mathbf{P}_{i_1, \dots, i_N}$. Cette propriété permet de prendre en compte les fortes non-linéarités des problèmes traités. Dans le cas d'une hypersurface B-spline, cela revient à :

$$N_{i_1, s_1}(\zeta_1) \times \dots \times N_{i_N, s_N}(\zeta_N) \neq 0 \text{ si } (\zeta_1, \dots, \zeta_N) \in [v_{i_1}^{(1)}, v_{i_1+s_1+1}^{(1)}[\times \dots \times [v_{i_N}^{(N)}, v_{i_N+s_N+1}^{(N)}[. \quad (3.8)$$

Pour plus d'informations sur le comportement et les propriétés fondamentales des entités NURBS, le lecteur est invité à se reporter à [138].

3.3 Génération du métamodèle basé sur les entités NURBS comme solution d'un problème d'optimisation

3.3.1 Description du problème

Les étapes d'utilisation d'un métamodèle basé sur les hypersurfaces NURBS sont, comme la plupart des méthodes de métamodélisation, l'étape *off-line* et l'étape *on-line*. La première correspond à la phase d'apprentissage et de calibration du métamodèle, durant laquelle la base de données \mathbf{Q} est construite et un sous-ensemble \mathbf{Q}_t de cette base est utilisé pour générer le métamodèle. La seconde étape correspond à l'utilisation du métamodèle, qui doit être la plus efficace et la moins demandeuse possible en ressources de calcul. Son objectif est, entre autres, de reproduire le comportement du modèle haute-fidélité initial avec un coût en ressources de calcul réduit, tout en assurant une précision comparable.

Le diagramme de flux illustré dans la Fig. 3.1 présente la stratégie complète de gé-

3.3 Génération du métamodèle basé sur les entités NURBS comme solution d'un problème d'optimisation

nération du métamodèle NURBS, comme solution d'un processus d'optimisation sous contraintes en trois étapes. Les étapes principales de cette stratégie sont rapidement décrites ci-dessous, avant d'être détaillées dans les sections suivantes :

1. la base de données est fournie en entrée du problème ainsi que les valeurs des paramètres entiers décrits dans la section 3.3.2. Dans ces travaux, cette base de données est évaluée à partir d'un échantillonnage de paramètres répartis dans l'espace selon une grille uniforme. La fonction objectif correspondant à une norme \mathcal{L}_2^2 , assimilable à un problème des moindres carrés, est également définie, tout comme les indicateurs de la précision du métamodèle, à savoir l'erreur relative moyenne $\bar{\varepsilon}_\alpha$ de la composante de sortie α avec la base de données de contrôle \mathbf{Q}_c et son coefficient de détermination linéaire de Poisson R_α^2 . La base de données de contrôle est évaluée de la même manière que la base d'entraînement \mathbf{Q}_t mais pour des coordonnées différentes ;
2. évaluation des CPs à partir des TP en considérant des entités B-splines : résolution analytique d'un problème aux moindres carrés. Plusieurs méthodes sont possibles : en considérant un terme de lissage ou pas, en résolvant le problème de manière globale ou par séparation des dimensions. Ces options, leurs avantages et leurs inconvénients, sont décrites plus en détail dans la section 3.4.1. Si les critères d'arrêt sont respectés, i.e., si $\bar{\varepsilon}_\alpha \leq \bar{\varepsilon}_{\text{obj}}$ et $R_\alpha^2 \geq R_{\text{obj}}^2$, cette première version du métamodèle basé sur des entités B-splines peut être employée ;
3. optimisation des vecteurs nodaux : résolution numérique d'un problème aux moindres carrés via un algorithme au gradient. Si les critères d'arrêt $\bar{\varepsilon}_\alpha \leq \bar{\varepsilon}_{\text{obj}}$ et $R_\alpha^2 \geq R_{\text{obj}}^2$ sont respectés, cette seconde version du métamodèle basé sur des entités B-splines peut être employée. Plus de détails sur cette optimisation sont donnés dans la section 3.4.2.1 ;
4. optimisation des poids associés aux CPs : résolution numérique d'un problème aux moindres carrés via un algorithme au gradient. Si les critères d'arrêt $\bar{\varepsilon}_\alpha \leq \bar{\varepsilon}_{\text{obj}}$ et $R_\alpha^2 \geq R_{\text{obj}}^2$ sont respectés, cette version du métamodèle basé sur des entités NURBS peut être employée. Plus de détails sur cette optimisation sont donnés dans la section 3.4.2.2 ;
5. le métamodèle optimisé, basé sur des entités B-splines ou NURBS selon le cas, est prêt à être utilisé dans le cadre d'un processus de conception, d'un processus d'optimisation, d'analyse de sensibilité, etc. Néanmoins, si les critères d'arrêt ne sont pas respectés, un enrichissement de la base de données d'entraînement devra être fait.

3.3.2 Initialisation du problème

Les entités NURBS sont pilotées par des variables de conception introduites dans l'Eq. (3.1) et l'Eq. (3.2) et séparées en deux catégories :

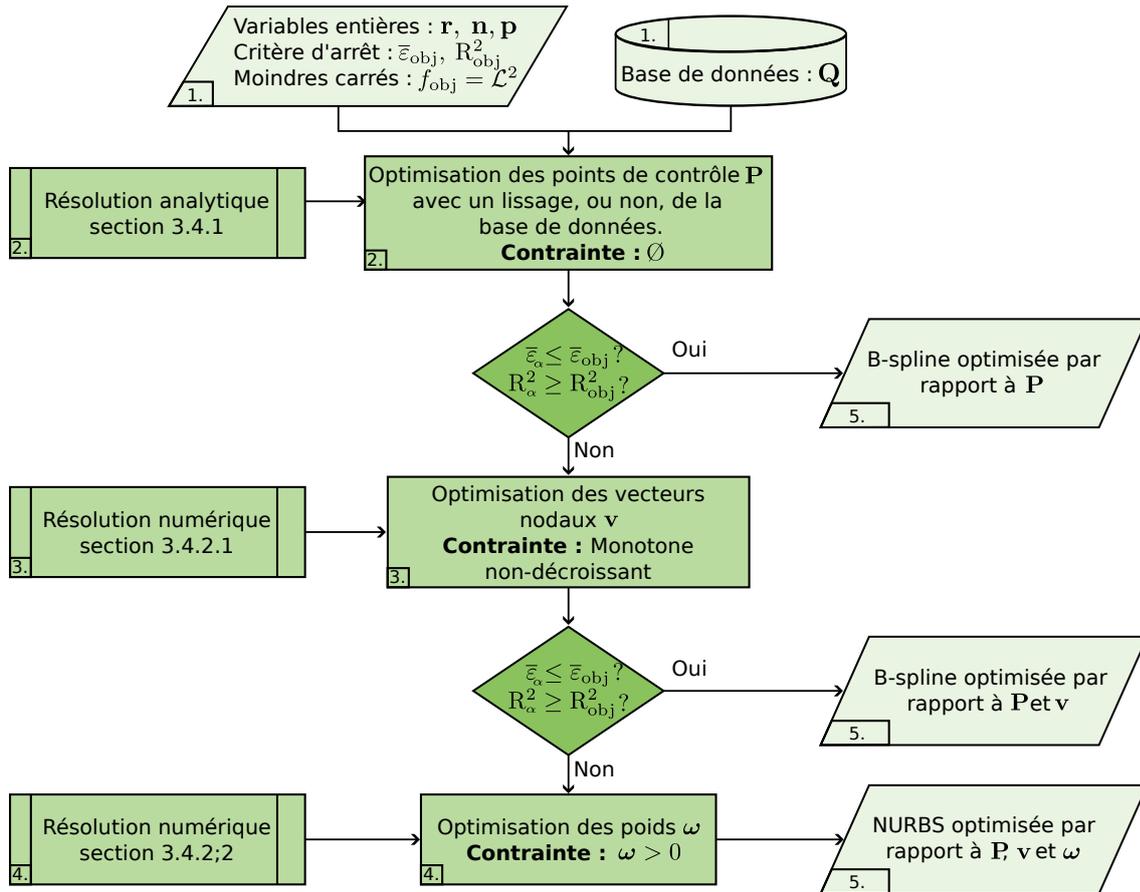


FIGURE 3.1 – Diagramme de flux de la génération du métamodèle comme solution d'un problème d'optimisation, basé sur des entités B-splines ou NURBS selon le cas. Les étapes d'optimisations ne sont pas détaillées par soucis de concision du diagramme.

- les variables entières, comprenant le nombre de CPs défini à partir de n_k , la longueur des vecteurs nodaux $m_k + 1$ et le degrés des polynômes de Bernstein s_k . A noter que la variable m_k dépend de n_k et s_k selon l'Eq. (3.4) ;
- les variables continues, comprenant les coordonnées des CPs et les poids ω associés à ces derniers, les valeurs non-triviales des vecteurs nodaux \mathbf{v}_k et les coordonnées sans dimension ζ_k auxquelles est évaluée l'hypersurface NURBS.

Dans le cadre de ce travail, seules les variables continues sont optimisées pour accroître l'efficacité du métamodèle, à l'exception des coordonnées paramétriques ζ_k qui sont liées aux variables d'entrée du métamodèle comme expliqué plus en détail dans la suite. Les variables entières sont déterminées à partir des règles métiers introduites dans [18, 38, 138] comme suit :

- les variables n_k sont directement déduites des variables r_k par le biais du rapport $\tau = \prod_{k=1}^N \tau_k = \prod_{k=1}^N \frac{n_k + 1}{r_k + 1}$. Pour obtenir des résultats cohérents, la valeur minimale acceptable de ce rapport est fixée à $\tau = \frac{1}{3}$ par [38]. Ainsi, de manière générale, il est fixé à $\tau = \frac{2}{3}$ pour un bon compromis entre précision du métamodèle et nombre de

3.3 Génération du métamodèle basé sur les entités NURBS comme solution d'un problème d'optimisation

CPs. Néanmoins, si le nombre de CPs à considérer est insuffisant, ce rapport peut être augmenté jusqu'à tendre vers 1 dans certains cas ;

- pour finir, les degrés s_k des polynômes de Bernstein dans la direction d'entrée $k \in [1, \dots, N]$ sont fixés à 2 ou 3 selon la non-linéarité du problème traité [18, 38, 138].

Avant de définir le problème d'optimisation qui va conduire à la génération du métamodèle basé sur des hypersurfaces NURBS, il est nécessaire d'initialiser les valeurs non-triviales des vecteurs nodaux et des poids. Le but de cette optimisation est de réduire les écarts entre le métamodèle et la base de données d'entraînement \mathbf{Q}_t . Durant ce processus, l'hypersurface NURBS est calculée aux coordonnées ζ_k correspondant aux valeurs des variables d'entrée, normées sur l'intervalle $[0, 1]$, utilisées pour générer les TPs de la base de données \mathbf{Q}_t . Lorsque les variables d'entrée du modèle d'étude u_k , ayant servi à générer les sorties stockées dans la base de données \mathbf{Q}_t , sont explicitement disponibles, le calcul de ζ_k revient à :

$$\zeta_{k,i} = \frac{u_{k,i} - u_k^{\text{LB}}}{u_k^{\text{UB}} - u_k^{\text{LB}}}, \quad (3.9)$$

avec $u_{k,i}$ correspondant à la i -ème valeur de la variable u_k , dans la direction $k \in [1, \dots, N]$ et u_k^{LB} , u_k^{UB} les bornes inférieure et supérieure de cette variable. Si u_k n'est pas disponible explicitement, des algorithmes de calcul de distance sont utilisés, comme l'algorithme *Centripetal* tiré de [108].

Selon la répartition des valeurs de ζ_k , l'initialisation des vecteurs nodaux peut se faire de deux manières. Si les valeurs ζ_k sont uniformément réparties sur leur intervalle de variation, les valeurs non-triviales des vecteurs nodaux le seront aussi. D'un autre côté, si cette répartition n'est pas uniforme, l'initialisation des valeurs non-triviales des vecteurs nodaux se fait par l'application de l'algorithme 1 de De Boor, repris par [61].

Pour finir, les poids ω_{i_1, \dots, i_N} associés à chaque CP peuvent être également initialisés de deux manières. La première consiste à employer des lois empiriques, énoncées notamment par Turner [178]. La seconde consiste à les initialiser à la même valeur (par exemple l'unité), ce qui revient à travailler avec des hypersurfaces B-splines (propriété de partition de l'unité Eq. (3.5)).

3.3.3 Énoncé du problème d'optimisation

La génération du métamodèle basé sur les hypersurfaces NURBS correspond à la résolution successive de problèmes d'optimisation sous contraintes. Plus précisément, il s'agit de problèmes assimilables à des problèmes aux moindres carrés sous contraintes, résolus analytiquement ou numériquement. Par abus de langage, le terme « aux moindres carrés » sera employé par la suite. Cette minimisation se fait via le calcul du gradient de la fonction objectif par rapport à la variable d'optimisation ξ . Dans des applications pratiques d'ingénierie, un problème d'optimisation est soumis à m_e contraintes d'égalité

Algorithme 1: Algorithme de calcul des vecteurs nodaux, inspiré de l'algorithme de De Boor [61].

Input: $\zeta_k; M; r_k; n_k; s_k$
Output: \mathbf{v}_k

```

1  $m_k \leftarrow n_k + s_k + 2$  ;
2  $\mathbf{v}_k[m_k] \leftarrow 0$  ;
3  $i \leftarrow 0$  ;
4  $\Delta \leftarrow \frac{r_k}{n_k - s_k + 1}$  ;
5 for  $j = 0 < s_k + 1$  do
6    $\mathbf{v}_{k,i} \leftarrow 0$  ;
7    $i \leftarrow i + 1$  ;
8 end
9 for  $j = 1 < n_k - s_k + 1$  do
10   $h \leftarrow \text{int}(j \times \Delta)$  ;
11   $\alpha \leftarrow j \times \Delta - h$  ;
12   $v_{k,i} \leftarrow (1 - \alpha)\zeta_{k,h} + \alpha\zeta_{k,h+1}$  ;
13   $i \leftarrow i + 1$  ;
14 end
15 for  $j = n_k + 1 < n_k + s_k + 2$  do
16   $\mathbf{v}_{k,i} \leftarrow 1$  ;
17   $i \leftarrow i + 1$  ;
18 end

```

de la forme $h_i(\boldsymbol{\xi}) = 0$, $i = 1, \dots, m_e$ et à m_i contraintes d'inégalité de la forme $g_j(\boldsymbol{\xi}) < 0$, $j = 1, \dots, m_i$ [129]. En outre, les intervalles de variation des variables d'optimisation $\boldsymbol{\xi}$ sont à considérer. Mathématiquement, un problème d'optimisation sous contraintes, appelé aussi CNLPP, est généralement formulé comme suit :

$$\begin{aligned}
& \min_{\boldsymbol{\xi}} f_{\text{obj}}(\boldsymbol{\xi}), \\
& \text{contraint par :} \\
& \begin{cases} h_i(\boldsymbol{\xi}) = 0, i = 1, \dots, m_e, \\ g_j(\boldsymbol{\xi}) < 0, j = 1, \dots, m_i, \\ \boldsymbol{\xi}_{\text{LB}} \leq \boldsymbol{\xi} \leq \boldsymbol{\xi}_{\text{UB}}, \end{cases} \tag{3.10}
\end{aligned}$$

avec $\boldsymbol{\xi}_{\text{LB}}$ et $\boldsymbol{\xi}_{\text{UB}}$ respectivement les bornes inférieure et supérieure de l'intervalle de variation de la variable d'optimisation $\boldsymbol{\xi}$. Pour plus d'informations, le lecteur est invité à se référer à [11, 18, 38, 129].

Pour le problème de génération du métamodèle basé sur les hypersurfaces NURBS, les variables d'optimisation sont les paramètres continus de l'entité NURBS définis plus haut, à savoir : (i) les coordonnées des CPs ; (ii) les composantes non triviales du vecteur nodal définies entre 0 et 1 dans chaque direction paramétrique du domaine de l'entité NURBS ; (iii) les poids associés aux CPs définis comme strictement positifs. Sur la base

de l'approche proposée dans [11,18,38,129], un processus d'optimisation en trois étapes est mis en œuvre dans ce travail, illustré dans les grandes lignes dans la Fig. 3.1. Chaque étape du processus illustré par le diagramme de flux de la Fig. 3.1 correspond à la résolution d'un CNLPP. Dans la première étape d'optimisation, les coordonnées optimales des CPs sont calculées grâce à une approche analytique détaillée dans la section 3.4.1. Au cours de cette étape, les composantes des vecteurs nodaux et des poids sont maintenues constantes à leur valeur d'initialisation. La solution de ce premier problème d'optimisation, est utilisée comme solution de départ pour l'optimisation suivante.

La seconde optimisation correspond à l'optimisation des valeurs non-triviales des vecteurs nodaux. Cette optimisation se fait par l'intermédiaire d'un algorithme au gradient. Les valeurs des coordonnées des CPs sont mises à jour à chaque itération et les poids sont maintenus constants. Le détail de cette étape est donné dans la section 3.4.2.1.

La dernière optimisation, correspondant à l'optimisation des poids, est également réalisée via un algorithme au gradient. Cette dernière étape utilise comme solution de départ les valeurs non triviales des vecteurs nodaux et les coordonnées des CPs fournies par l'étape précédente. Cette optimisation séquentielle est détaillée dans la section 3.4.2.2.

Pour les deux dernières phases d'optimisation, qui sont des étapes purement numériques, le calcul du gradient analytique est couplé à un algorithme au gradient basé sur la méthode *Globally Convergent Method of Moving Asymptotes* (GC-MMA) [174] généralement employé dans le domaine de l'optimisation topologique [125]. Dans cette thèse l'objet du processus d'optimisation est un métamodèle et non une topologie, mais le grand nombre de variables d'optimisation inhérent aux entités NURBS à considérer (jusqu'à 5×10^4), a justifié l'emploi de cette méthode. En outre, une version améliorée de cet algorithme étant développée au sein de l'I2M [125], il n'est pas considéré comme une boîte noire, ce qui facilite entre autre la répétabilité des opérations sur différentes machines.

Pour chaque étape d'optimisation du processus de génération du métamodèle, la même fonction objectif est employée, exprimée par :

$$\begin{aligned}
 f_{\text{obj}} &= \frac{1}{f_0} \sum_{\alpha=1}^M f_{\text{obj},\alpha} \\
 &= \frac{1}{f_0} \sum_{\alpha=1}^M \sum_{i=1}^{n_{\text{TP}}} [h_{\alpha}(\zeta_{1,i}, \dots, \zeta_{N,i}) - q_{\alpha,i}]^2,
 \end{aligned} \tag{3.11}$$

avec f_0 la valeur de la fonction objectif avant optimisation qui permet une normalisation du problème (de cette manière la valeur maximale sera égale à l'unité et celle minimale sera théoriquement zéro), α la composante générique du vecteur des sorties considérée dans l'espace de dimensions \mathbb{R}^M , n_{TP} le nombre de TPs, $h_{\alpha}(\zeta_{1,i}, \dots, \zeta_{N,i})$ le i -ème point de l'hypersurface dans la direction de sortie α évalué aux coordonnées paramétriques $(\zeta_{1,i}, \dots, \zeta_{N,i})$ et $q_{\alpha,i}$ le point correspondant de la base de donnée génératrice évalué aux mêmes coordonnées paramétriques.

En introduisant les indices linéaires définis comme :

$$\begin{cases} \kappa_{\text{CP}} = 1 + \sum_{j=1}^N \left(i_j \prod_{k=1}^{j-1} (n_k + 1) \right), i_j = 0, \dots, n_j, \\ \kappa_{\text{TP}} = 1 + \sum_{j=1}^N \left(i_j \prod_{k=1}^{j-1} (r_k + 1) \right), i_j = 0, \dots, r_j, \end{cases} \quad (3.12)$$

avec $\kappa_{\text{CP}} = 1, \dots, n_{\text{CP}}$ et $\kappa_{\text{TP}} = 1, \dots, n_{\text{TP}}$, l'Eq. (3.11) peut s'écrire en forme plus compacte comme :

$$f_{\text{obj}} = \frac{1}{f_0} \sum_{\alpha=1}^M (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha), \quad (3.13)$$

où $\mathbf{h}_\alpha = \mathbf{N}\mathbf{p}_\alpha$. Dans la formule ci-dessus, le vecteur $\mathbf{q}_\alpha \in \mathbb{R}^{n_{\text{TP}}}$ collecte la composante α de chaque TP évaluée pour les diverses combinaisons des variables d'entrée, tandis que le vecteur $\mathbf{p}_\alpha \in \mathbb{R}^{n_{\text{CP}}}$ collecte la composante α des CPs de l'hypersurface NURBS.

La matrice $\mathbf{N} \in \mathbb{R}^{n_{\text{TP}} \times n_{\text{CP}}}$ est définie à partir de l'équation suivante :

$$N_{\kappa_{\text{TP}}, \kappa_{\text{CP}}} = N_{\kappa_{\text{TP}}, \kappa_{\text{CP}}}^* = N_{i_1, s_1}(\zeta_{1, \kappa_{\text{TP}}}) \times \dots \times N_{i_N, s_N}(\zeta_{N, \kappa_{\text{TP}}}), \quad (3.14)$$

dont le passage de i à κ_{CP} peut être déduit de l'Eq. (3.12). La formule précédente est valable dans le cas d'une hypersurface B-spline. Pour une hypersurface NURBS il faut remplacer les fonctions de base $N_{-, -}$ par les fonctions $R_{-, -}$ de l'Eq. (3.2).

Lorsque la base de données \mathbf{Q} comporte du bruit ou de fortes oscillations, il est nécessaire d'intégrer à la génération du métamodèle un terme de lissage J_α . Le but de ce terme est alors de réduire artificiellement l'influence des perturbations inhérentes à la base de données sur le comportement du métamodèle résultant et/ou de réduire/éviter le phénomène d'*overfitting*. En considérant ce terme, la fonction objectif de l'Eq. (3.11) devient :

$$f_{\text{obj, liss}} = \frac{1}{f_0} \sum_{\alpha=1}^M (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha) + \lambda J_\alpha, \quad (3.15)$$

avec J_α exprimé sous sa forme générale par [96, 183] :

$$J_\alpha = \sum_{d_1 + \dots + d_N = d} \binom{d}{d_1, \dots, d_N} \int_0^1 \dots \int_0^1 \left(\frac{\partial^d h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i}) - q_{\alpha,i}}{\partial \zeta_1^{d_1}, \dots, \partial \zeta_N^{d_N}} \right)^2 d\zeta_1, \dots, d\zeta_N \quad (3.16)$$

où $d = \sum_k^N d_k$ est l'ordre maximal de dérivation considéré, fixé à $d = 2$ tout au long de ce travail, et d_i les ordres de dérivation partielle. En utilisant l'indice linéaire de l'Eq. (3.12), le terme de lissage J_α peut être exprimé sous forme matricielle comme suit :

$$J_\alpha = \mathbf{p}_\alpha^T \mathbf{E} \mathbf{p}_\alpha, \text{ avec } \mathbf{E} \in \mathbb{R}^{n_{\text{CP}} \times n_{\text{CP}}}. \quad (3.17)$$

A titre d'exemple, dans le cas d'une hypersurface B-spline, les termes de la matrice \mathbf{E}

s'écrivent :

$$E_{\kappa_{\text{CP},1}, \kappa_{\text{CP},2}} = \sum_{d_1 + \dots + d_N = d} \binom{d}{d_1, \dots, d_N} \int_0^1 \dots \int_0^1 \int_0^1 \dots \int_0^1 \frac{\partial^d N_{\kappa_{\text{CP},1}}^*(\zeta_1, \dots, \zeta_N)}{\partial \zeta_1^{d_1} \dots \partial \zeta_N^{d_N}} \frac{\partial^d N_{\kappa_{\text{CP},2}}^*(\zeta_1, \dots, \zeta_N)}{\partial \zeta_1^{d_1} \dots \partial \zeta_N^{d_N}} d\zeta_1 \dots d\zeta_N, \quad (3.18)$$

Pour ce qui du poids λ , il est calculé arbitrairement dans [18, 56] pour un cas de surface. En généralisant son expression, on obtient :

$$\lambda = \frac{\|\mathbf{N}^T \mathbf{N}\|_2}{\|\mathbf{E}\|_2}, \quad (3.19)$$

où $\|\cdot\|_2$ correspond à la norme \mathcal{L}_2 exprimée par $\|\mathbf{K}\|_2 = \left(\sum_{i,j} K_{i,j}^2\right)^{\frac{1}{2}}$.

3.4 Mise en place du processus d'optimisation

Cette section présente en détail les différentes étapes d'optimisation inhérentes à chacune des variables d'optimisation. Ces dernières peuvent être regroupées dans trois vecteurs ξ_j , avec $j = 1, 2, 3$, optimisés par trois CNLPP successifs :

- $\xi_1 \in \mathbb{R}^{M n_{\text{CP}}}$ rassemble les valeurs des coordonnées des CPs sous la forme $\xi_1^T = \{\mathbf{p}_0^T, \dots, \mathbf{p}_\alpha^T, \dots, \mathbf{p}_M^T\}$;
- $\xi_2 \in \mathbb{R}^{n_{\text{knot}}}$ avec $n_{\text{knot}} = \sum_{k=1}^N m_k + 1$ est relié aux valeurs non triviales des vecteurs nodaux, monotones non-décroissantes, notées $v_{i_k}, i_k \in [s_k + 1, m_k - s_k - 1]$ avec $k = 1, \dots, N$;
- $\xi_3 \in \mathbb{R}^{n_{\text{CP}}}$ rassemble l'ensemble des valeurs des poids ω_{i_1, \dots, i_N} associés à chaque CP.

3.4.1 Résolution du CNLPP pour l'évaluation des coordonnées des CPs

3.4.1.1 Formulation générale

Les coordonnées optimales des CPs sont déterminées par résolution du CNLPP défini à la section 3.3.3 par rapport au vecteur ξ_1 . Pour cette première étape, les poids prennent la même valeur permettant ainsi d'employer une hypersurface B-spline plutôt qu'une NURBS. Ainsi, en considérant la formulation de l'Eq. (3.13) et l'approche globale proposée par [11], la solution de ce CNLPP, pour la composante de sortie α , peut s'exprimer analytiquement par :

$$\begin{aligned} \min_{\mathbf{p}_\alpha} (f_{\text{obj}, \alpha}) &\Rightarrow \frac{\partial}{\partial p_{\alpha, \kappa_{\text{CP}}}} (\mathbf{N} \mathbf{p}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{N} \mathbf{p}_\alpha - \mathbf{q}_\alpha) = 0 \\ &\Leftrightarrow 2 \mathbf{N}^T (\mathbf{N} \mathbf{p}_\alpha - \mathbf{q}_\alpha) = 0, \\ &\Leftrightarrow \mathbf{p}_\alpha = (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \mathbf{q}_\alpha. \end{aligned} \quad (3.20)$$

Pour plus de clarté dans la formulation matricielle, les structures respectives des vecteurs \mathbf{q}_α et \mathbf{p}_α ainsi que de la matrice \mathbf{N} sont représentées ci-dessous, en considérant les indices linéaires définis par l'Eq. (3.12) :

$$\mathbf{p}_\alpha = \begin{bmatrix} p_{\alpha,1} \\ \vdots \\ p_{\alpha,\kappa_{CP}} \\ \vdots \\ p_{\alpha,n_{CP}} \end{bmatrix}, \quad \mathbf{q}_\alpha = \begin{bmatrix} q_{\alpha,1} \\ \vdots \\ q_{\alpha,\kappa_{TP}} \\ \vdots \\ q_{\alpha,n_{TP}} \end{bmatrix}, \quad (3.21)$$

$$\mathbf{N} = \begin{bmatrix} N_{0,s_1}(\zeta_{1,0}) \times \cdots \times N_{0,s_N}(\zeta_{N,0}) & \cdots & N_{n_1,s_1}(\zeta_{1,0}) \times \cdots \times N_{n_N,s_N}(\zeta_{N,0}) \\ \vdots & \ddots & \vdots \\ N_{0,s_1}(\zeta_{1,n_{TP}}) \times \cdots \times N_{0,s_N}(\zeta_{N,n_{TP}}) & \cdots & N_{n_1,s_1}(\zeta_{1,n_{TP}}) \times \cdots \times N_{n_N,s_N}(\zeta_{N,n_{TP}}) \end{bmatrix}. \quad (3.22)$$

En fonction des non-linéarités présentes dans la base de données \mathbf{Q} , une réduction de leur influence sur le calcul des coordonnées des CPs peut être nécessaire via l'introduction du terme de lissage J_α (Eq. (3.15)). En se basant sur le cas surfacique de [18] et sur la formulation matricielle de J_α de l'Eq. (3.17), la résolution analytique du CNLPP relié aux coordonnées des CPs est donnée par :

$$\begin{aligned} \min_{\mathbf{p}_\alpha} (f_{\text{obj,liss},\alpha}) &\Rightarrow \frac{\partial}{\partial p_{\alpha,\kappa_{CP}}} \left((\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha) + \lambda J_\alpha \right) = 0, \\ &\Leftrightarrow \frac{\partial}{\partial p_{\alpha,\kappa_{CP}}} \left((\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha) + \lambda \mathbf{p}_\alpha^T \mathbf{E} \mathbf{p}_\alpha \right) = 0, \\ &\Leftrightarrow 2\mathbf{N}^T (\mathbf{N}\mathbf{p}_\alpha - \mathbf{q}_\alpha) + \lambda 2\mathbf{E} \mathbf{p}_\alpha = 0, \\ &\Leftrightarrow \mathbf{p}_\alpha = \left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right)^{-1} \mathbf{N}^T \mathbf{q}_\alpha. \end{aligned} \quad (3.23)$$

Le diagramme présenté dans la Fig. 3.2 illustre la mise en place numérique de cette formulation selon les étapes suivantes :

1. la base de données $\mathbf{Q} = \{\mathbf{q}_1, \dots, \mathbf{q}_M\}$ de dimension $M \times n_{TP}$ est fournie en entrée du problème et le compteur des dimensions de sortie est initialisé à $\alpha = 0$;
2. le vecteur \mathbf{q}_α est défini ;
3. les coordonnées des CPs de la composante de sortie α sont calculées via l'Eq. (3.20) ou l'Eq. (3.23), respectivement si le terme de lissage J_α est pris en compte ou non ;
4. le compteur α est incrémenté de 1 s'il est inférieur à M ;
5. l'algorithme renvoie la matrice $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_M\}$ de dimension $M \times n_{CP}$ contenant les coordonnées des CPs.

Bien que cette formulation permette une résolution rapide du problème d'optimisation par manipulation de vecteurs et matrices, les tailles de ces derniers peuvent facilement

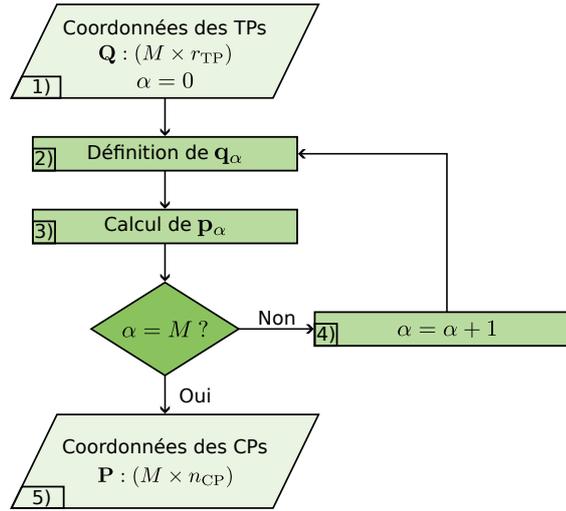


FIGURE 3.2 – Résolution du problème aux moindres carrés pour la détermination des coordonnées des *Control Points* via la formulation globale.

atteindre les limites numériques de l'ordinateur : lorsque le problème monte en dimension, la quantité de donnée croît exponentiellement [72]. Par exemple, considérons un cas MIMO avec $N = 3$, $M = 3$, $r_k = 20$ et $\tau = \frac{2}{3}$, soit $\tau_k = \sqrt[3]{\frac{2}{3}}$, i.e., $n_k = 17$. La matrice \mathbf{N} est alors constituée de $n_{\text{TP}} \times n_{\text{CP}} = 33 \times 10^6$ éléments, occupant 250 Mo de mémoire (matrice construite via la librairie numpy de Python). Sur un ordinateur classique, si le nombre de dimensions augmente encore, cette matrice peut être difficile voire impossible à manipuler. Pour s'affranchir de cette limitation, la solution, proposée initialement par [11] et décrite ci-après, est d'approximer le problème global d'approximation d'hypersurface en N sous-problèmes d'approximation de courbe.

3.4.1.2 Formulation du problème d'approximation d'hypersurface par séparation de variables

Les coordonnées optimales des CPs sont déterminées par la résolution du CNLPP définie à la section 3.3.3. Dans l'optique de réduire les coûts de calculs, [9, 11] propose de séparer le problème selon ses dimensions d'entrée et de sortie. En se basant sur l'Eq. (3.20), la formulation récursive de ces problèmes est donnée par :

$$\min_{\mathbf{p}_\alpha} (f_{\text{obj}, \alpha}) \Rightarrow \mathbf{P}_\alpha^k = \left(\mathbf{N}_k^T \mathbf{N}_k \right)^{-1} \mathbf{N}_k^T \mathbf{Q}_\alpha^{k-1}, \text{ avec } k = 1, \dots, N. \quad (3.24)$$

Lors de la première récursion, $k = 1$, l'array $\mathbf{Q}_\alpha^{k-1} = \mathbf{Q}_\alpha$ est de taille $(r_1 + 1) \times \dots \times (r_N + 1)$. Lors des récursions suivantes, la taille de sa dimension $k - 1$ est réduite de $(r_{k-1} + 1)$ à $(n_{k-1} + 1)$ jusqu'à tendre vers sa dimension finale de $(n_1 + 1) \times \dots \times (n_{N-1} + 1) \times (r_N + 1)$, avec $k = N$. En même temps, la taille de l'array \mathbf{P}_α^{k-1} est initialisée à $(n_1 + 1) \times (r_2 + 1) \times \dots \times (r_N + 1)$. Lors des récursions suivantes, la taille de sa dimension k est réduite de $(r_k + 1)$ à $(n_k + 1)$ jusqu'à tendre vers sa dimension finale de $(n_1 + 1) \times \dots \times (n_N + 1)$.

Quant à la matrice $\mathbf{N}_k \in \mathbb{R}^{(r_k+1) \times (n_k+1)}$ qui rassemble les polynômes de Bernstein dans la dimension k , elle est construite via la formule suivante :

$$N_{k,i_k,j_k}^* = N_{j_k,p_k}(\zeta_{i_k}), \quad (3.25)$$

avec $i_k = 0, \dots, r_k$ et $j_k = 0, \dots, n_k$. Numériquement, la matrice \mathbf{N}_k est construite comme suit :

$$\mathbf{N}_k = \begin{bmatrix} N_{0,s_k}(\zeta_0) & \cdots & N_{n_k+1,s_k}(\zeta_0) \\ \vdots & \ddots & \vdots \\ N_{0,s_k}(\zeta_{r_k}) & \cdots & N_{n_k+1,s_k}(\zeta_{r_k}) \end{bmatrix} \quad (3.26)$$

En considérant le terme de lissage, le calcul des coordonnées des CPs de l'Eq. (3.24) devient :

$$\min_{\mathbf{P}_\alpha} (f_{\text{obj,liss},\alpha}) \Rightarrow \mathbf{P}_\alpha^k = \left(\mathbf{N}_k^T \mathbf{N}_k + \lambda \mathbf{E}_k \right)^{-1} \mathbf{N}_k^T \mathbf{Q}_\alpha^{k-1}, \text{ avec } k = 1, \dots, N, \quad (3.27)$$

avec la matrice \mathbf{E}_k construite via :

$$E_{k,i_k,j_k} = \int_0^1 \frac{\partial^2 N_{i_k,s_k}^*(\zeta_k)}{\partial \zeta_k^2} \frac{\partial^2 N_{j_k,s_k}^*(\zeta_k)}{\partial \zeta_k^2} d\zeta_k \quad (3.28)$$

où $i_k, j_k = 1, \dots, n_k$. A partir de l'Eq. (3.28), le terme λ peut être reformulé par :

$$\lambda = \max_k (\lambda_k) = \max_k \left(\frac{\|\mathbf{N}_k^T \mathbf{N}_k\|}{\|\mathbf{E}_k\|} \right). \quad (3.29)$$

Lors de la formulation du problème d'approximation d'hypersurface par séparation de variables, non seulement l'évaluation des coordonnées des CPs doit être adaptée, mais également le calcul du terme de lissage J_α de la fonction objectif de l'Eq. (3.15). Ce terme initialement exprimé en fonction de \mathbf{E} , est décomposé selon :

$$J_\alpha = \sum_{k=1}^N \|\mathbf{J}_{\alpha,k}\|_s, \quad (3.30)$$

où $\|\cdot\|_s$ est la norme \mathcal{L}_s d'une matrice exprimée par $\|\mathbf{A}\|_s = \left(\sum_{i=1}^n \sum_{j=1}^m A_{i,j}^2 \right)^{\frac{1}{s}}$ avec \mathbf{A} une matrice quelconque. Dans la suite la valeur $s = 8$ est prise en compte pour le calcul de la norme \mathcal{L}_s . Quant à la matrice de lissage $\mathbf{J}_{\alpha,k} \in \mathbb{R}^{(n_k+1) \times (n_k+1)}$ dans la direction d'entrée k , elle s'exprime par :

$$\mathbf{J}_{\alpha,k} = \mathbf{P}_{\alpha,k}^{*\top} \mathbf{E}_k \mathbf{P}_{\alpha,k}^*, \quad (3.31)$$

où $\mathbf{P}_{\alpha,k}^* \in \mathbb{R}^{(n_k+1) \times n_{\text{CP}}^*}$ avec $n_{\text{CP}}^* = \prod_{k^* \neq k}^N (n_{k^*} + 1)$. Dans l'Eq. (3.31), la matrice $\mathbf{P}_{\alpha,k}^*$ est

construite à partir de l'array \mathbf{P}_α par :

$$P_{\alpha,k,i_k,\kappa_{\text{CP}}^*}^* = P_{\alpha,k,i_1,\dots,i_k,\dots,i_N} \quad (3.32)$$

avec $i_k = 0, \dots, n_k$ et $\kappa_{\text{CP}}^* = 0, \dots, n_{\text{CP}}^*$ donné par l'Eq. (3.12) sans considérer les termes de la dimension k .

Le diagramme présenté dans la Fig. 3.3 illustre la mise en place numérique de cette formulation selon les étapes suivantes :

1. la base de données $\mathbf{Q} = \{\mathbf{Q}_1, \dots, \mathbf{Q}_M\}$ est fournie en entrée du problème, et les compteurs des espaces d'entrée et de sortie sont initialisés à $k = 0$ et $\alpha = 0$, respectivement ;
2. l'array $\mathbf{Q}_{\alpha,k-1}$ est défini. Sa taille initiale, i.e., $k = 1$, est de $(r_1 + 1) \times \dots \times (r_N + 1)$;
3. l'array $\mathbf{P}_{\alpha,k}$ est calculé à partir de l'Eq. (3.24) ou de l'Eq. (3.27) selon si le terme de lissage est pris en compte. Lors de la première itération, sa dimension est de $(n_1 + 1) \times (r_2 + 1) \times \dots \times (r_N + 1)$;
4. si le compteur $k < N$, alors $k = k + 1$ et la taille $(r_k + 1)$ est réduite à $(n_k + 1)$. Le processus recommence de l'étape 3) ;
5. si le compteur $k = N$, alors l'array $\mathbf{P}_{\alpha,k} = \mathbf{P}_\alpha$ de taille $(n_1 + 1) \times \dots \times (n_N + 1)$. Quant à l'array $\mathbf{Q}_{\alpha,k-1}$, sa taille finale est de $(n_1 + 1) \times \dots \times (n_{N-1} + 1) \times (r_N + 1)$;
6. si le compteur $\alpha < M$, alors $\alpha = \alpha + 1$ et la boucle récursive de calcul recommence depuis l'étape 2) ;
7. si le compteur $\alpha = M$, alors la totalité des coordonnées des CPs sont calculées et fournies en sortie de l'algorithme sous la forme $\mathbf{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_M\}$.

3.4.1.3 Choix de la formulation

Dans une situation où le terme de lissage n'est pas pris en compte, les deux formulations précédentes sont équivalentes [11]. La différence réside seulement dans la taille des vecteurs et matrices à manipuler lors de la résolution du CNLPP optimisant les coordonnées des CPs. Les termes qui ne sont pas pris en compte dans la formulation des matrices \mathbf{N}_k par rapport à la matrice \mathbf{N} sont implicitement recalculés lors de la construction récursive de \mathbf{P} .

Néanmoins, la formulation générale du problème d'approximation d'hypersurface et celle approximée par séparation de variable diffèrent lors de la prise en compte du terme de lissage. Pour illustrer cet écart et évaluer son impact, considérons le problème suivant :

$$z = 2a \cos(\theta)^2 \sin(\phi) + \Phi, \quad (3.33)$$

avec $\theta \in [0, 2\pi]$, $\phi \in [10, 20]$ et $a = 0.8$. Il s'agit d'un modèle analytique mathématique qui décrit une surface évoluant dans un espace cartésien classique, i.e., $N = 2$ et $M = 3$. Pour

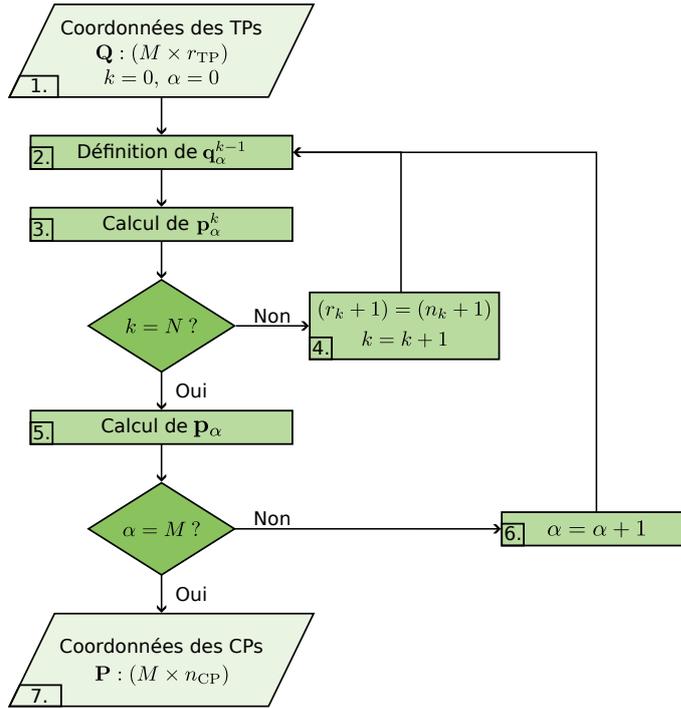


FIGURE 3.3 – Résolution du problème aux moindres carrés pour la détermination des coordonnées des *Control Points* via la formulation par séparation des dimensions.

TABLE 3.1 – Caractéristiques communes aux métamodèles employés pour comparer les approches globale et par séparation des dimensions.

Variable	s_k	r_k	n_k
θ	3	26	17
ϕ	3	32	21

que le terme de lissage ait un rôle, du bruit est artificiellement introduit via la variable Φ qui suit une loi uniforme centrée en 0 définie sur l'intervalle $\mathcal{U}[-0.05, 0.05]$. Les rapports τ_k sont fixés à $\tau_k = \frac{2}{3}$. Afin de comparer les approches globale et par séparation des dimensions, une surface de réponse est construite pour chaque situation. Leur caractéristiques communes sont regroupées dans le Tab. 3.1 avec notamment r_k le nombre de point cible de la base d'entraînement dans la direction k et n_k le nombre de point de CPs dans la direction k .

Les indicateurs suivants sont employés pour évaluer les précisions respectives des métamodèles, par rapport à la base de données d'entraînement lisse, notée $\mathbf{Q}_{t,\text{liss}}$:

- le coefficient de détermination linéaire de Poisson R^2 est calculé pour chaque dimension de sortie α par :

$$R_\alpha^2 = 1 - \frac{\sum_{i=1}^{n_{\text{TP}}} (q_{\alpha,i} - h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i}))^2}{(q_{\alpha,i} - \bar{q}_\alpha)^2}, \quad (3.34)$$

avec \bar{q}_α la valeur moyenne des TPs dans la direction α ;

— l'erreur relative moyenne¹ pour chaque sortie α est exprimée par :

$$\bar{\varepsilon}_\alpha = \frac{1}{n_{\text{TP}}} \sum_{i=1}^{n_{\text{TP}}} \frac{|q_{\alpha,i} - h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i})|}{|q_{\alpha,i}|}, \quad (3.35)$$

— la norme \mathcal{L}_1 entre les métamodèles construits via les formulations globale $\mathbf{h}_{\alpha,\text{glob}}$ et par séparation de variables $\mathbf{h}_{\alpha,\text{sep}}$ s'exprime par :

$$\mathbf{e}_{M,\alpha} = |\mathbf{h}_{\alpha,\text{sep}} - \mathbf{h}_{\alpha,\text{glob}}|. \quad (3.36)$$

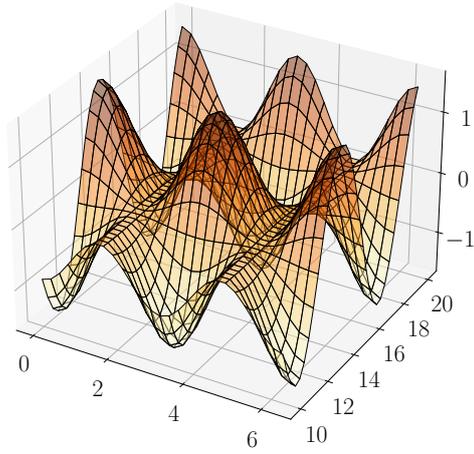
La Fig. 3.4a correspond à la base de données initiale générée à partir de l'Eq. (3.33) sans considérer le bruit introduit par Φ , là où la Fig. 3.4b en tient compte. La Fig. 3.4e trace les points générés par le premier métamodèle en fonction des points calculés par le second. Pour finir, les courbes de niveaux de la Fig. 3.4f illustrent les écarts entre les métamodèles calculés par l'Eq. (3.36).

Afin de minimiser les biais pouvant être introduits par le comportement aléatoire de la loi uniforme Φ , les métamodèles associés à chaque formulations sont évalués sur 200 bases de données $\mathbf{Q}_{t,\text{liss}}$. Une moyenne des résultats est ensuite faite, avant de les comparer via les indicateurs $\bar{\varepsilon}$ et R^2 . Il ressort alors un coefficient moyen R^2 de $R_z^2 = 0.998862$, pour la dimension la plus critique, $\alpha = z$. Cette correspondance peut également être visualisée sur la Fig. 3.4e. En outre, les erreurs relatives moyennes suivant la composante de sortie $\alpha = z$ sont de $\bar{\varepsilon}_{\text{sep}} = 0.6 \%$ et $\bar{\varepsilon}_{\text{glob}} = 0.7 \%$. Compte tenu de ces résultats, nous pouvons conclure que la formulation globale et par séparation de dimensions fournissent des résultats équivalents. Cependant, dans la mesure où la deuxième formulation nécessite moins de ressource, elle sera utilisée dans la suite de ce travail.

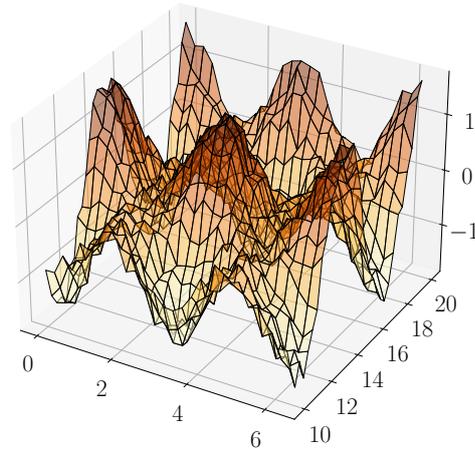
3.4.2 Optimisation numérique

La résolution du premier CNLPP pour calculer les valeurs optimales des coordonnées des CPs s'est faite analytiquement. Les deux autres CNLPP pour optimiser respectivement les valeurs non-triviales des vecteur nodaux et les poids n'ont pas de solution analytique directe. Par conséquent, la résolution de ces derniers se fait via l'application d'un algorithme au gradient en considérant les gradients analytiques des fonctions objectifs par rapport à ξ_2 et ξ_3 . Pour ces deux problèmes d'optimisation, parmi les différents algorithmes au gradient disponibles dans les bibliothèques Python, deux stratégies peuvent être utilisées : l'algorithme GC-MMA [174] et l'algorithme *Sequential Least Squares Programming* (SLSQP) [100]. Ces deux algorithmes sont assez robustes et peuvent converger même si le point initial de l'optimisation se trouve dans une zone non faisable du domaine d'optimisation. Dans ce travail, l'algorithme GC-MMA est préféré parce qu'il montre une

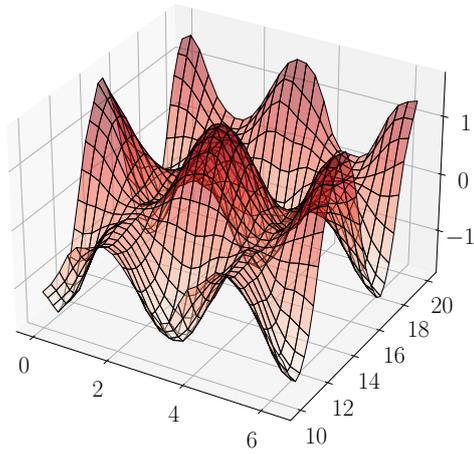
1. La valeur de l'erreur relative moyenne sera systématiquement exprimée en pourcentage pour faciliter sa lecture.



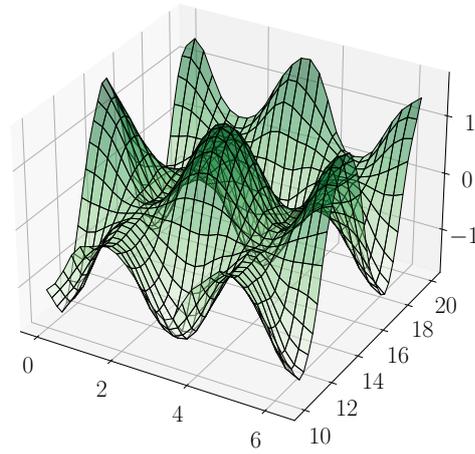
(a) Base de données initiale.



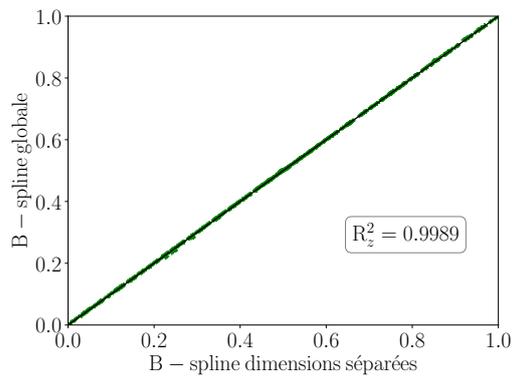
(b) Base de données bruitée.



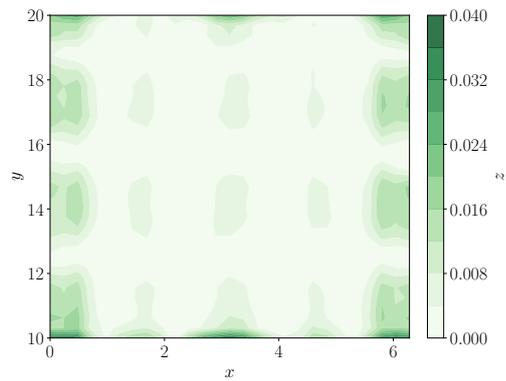
(c) Métamodèle construit par formulation globale avec lissage.



(d) Métamodèle construit par formulation à séparation de variables avec lissage.



(e) Graphique traçant les résultats fournis par la méthode globale vs. les résultats fournis par la méthode par séparation de dimension.



(f) Courbes de niveaux des écarts entre les deux métamodèles.

FIGURE 3.4 – Comparaison des métamodèles issus des deux formulations, globale et par séparation des dimensions, sur un cas analytique bruité.

TABLE 3.2 – Paramètres de l'algorithme *Globally Convergent Method of Moving Asymptotes*.

Paramètre	Valeur
<i>move</i>	0.1
<i>albefa</i>	0.1
Critère d'arrêt	Valeur
Nombre maximal d'évaluation de f_{obj}	10000
Nombre maximal d'itération	100
Tolérance sur f_{obj}	10^{-8}
Tolérance sur les variables optimisées	10^{-8}
Tolérance sur la norme Karush-Kuhn-Tucker	10^{-6}

plus grande robustesse, notamment via les apports de [125], et une meilleure efficacité lorsque le nombre de variables d'optimisation est élevé². Pour tous les cas traités dans ce travail, les paramètres de l'algorithme GC-MMA sont détaillés dans le Tab. 3.2. Pour plus de détails sur le développement de l'algorithme GC-MMA, se référer à [129, 174].

3.4.2.1 Optimisation des vecteurs nodaux

Cette phase d'optimisation, tout comme la première, se base sur des entités B-spline. Son rôle est d'optimiser les valeurs non-triviales des vecteurs nodaux qui doivent respecter la contrainte de non-décroissance monotone sur l'intervalle $[0, 1]$. Pour permettre une optimisation efficace de ces variables, le changement de variable suivant est opéré :

$$v_{i_k} = \beta_{i_k} v_{i_k-1} + (1 - \beta_{i_k}) v_{i_k+1} \quad (3.37)$$

avec v_{i_k} le i -ème nœud non trivial du vecteur nodal dans la direction d'entrée k et $\beta_{i_k} \in [0, 1]$ la variable de conception considérée pour l'optimisation. Le principal avantage dans ce changement de variable est le respect intrinsèque de la contrainte de v_{i_k} . La mise en place de ce second processus d'optimisation se fait par le calcul du gradient de la fonction objectif par rapport aux coefficients β_{i_k} contenus dans le vecteur :

$$\boldsymbol{\xi}_2 = \{\beta_{p_1+1}, \dots, \beta_{m_1-p_1-1}, \dots, \beta_{i_k}, \dots, \beta_{p_N+1}, \dots, \beta_{m_N-p_N-1}\}, \boldsymbol{\xi}_2 \in \mathbb{R}^{n_{\text{knot}}}. \quad (3.38)$$

L'initialisation du vecteur $\boldsymbol{\xi}_2$ se fait via l'Eq. (3.37) en considérant les valeurs initiales des vecteurs nodaux déterminées par l'algorithme 1, coïncidant avec la densité de répartition des paramètres u_k de la base de données \mathbf{Q}_t .

2. L'algorithme SLSQP est efficace pour un nombre de variables à optimiser inférieur à 2000.

Résolution du CNLPP sans lissage

La résolution du CNLPP associé à ξ_2 , sans prendre en compte le terme de lissage, revient à minimiser la valeur de la fonction objectif de l'Eq. (3.11) :

$$\min_{\xi_2} (f_{\text{obj}}) \Rightarrow \frac{\partial f_{\text{obj}}}{\partial \xi_{2,j}} = 0, \text{ avec } j = 0, \dots, n_{\text{knot}}. \quad (3.39)$$

La résolution de cette équation ne comportant pas de solution analytique explicite, l'emploi d'un algorithme au gradient est nécessaire. Il permet, par itérations successives, d'approcher la solution du problème de l'Eq. (3.39). Le calcul du gradient de f_{obj} par rapport au coefficient β_{i_l} est donnée par :

$$\frac{\partial f_{\text{obj}}}{\partial \beta_{i_l}} = \sum_{\alpha=1}^M \left(\sum_{i=1}^{n_{\text{TP}}} 2 (h_{\alpha}(\zeta_{1,i}, \dots, \zeta_{N,i}) - q_{i,\alpha}) \frac{\partial h_{\alpha}}{\partial v_{i_l}} (v_{i-1} - v_{i+1}) \right), \quad (3.40)$$

ou en utilisant une notation compacte :

$$\frac{\partial f_{\text{obj}}}{\partial \beta_{i_l}} = \sum_{\alpha=1}^M 2 (\mathbf{N}\mathbf{p}_{\alpha} - \mathbf{q}_{\alpha})^{\text{T}} \frac{\partial \mathbf{h}_{\alpha}}{\partial v_{i_l}} (v_{i-1} - v_{i+1}), \quad (3.41)$$

avec $\mathbf{h}_{\alpha} = \mathbf{N}\mathbf{p}_{\alpha}$. Le terme critique de cette expression est $\frac{\partial \mathbf{h}_{\alpha}}{\partial v_{i_l}}$, exprimé par :

$$\frac{\partial \mathbf{h}_{\alpha}}{\partial v_{i_l}} = \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_{\alpha} + \mathbf{N} \frac{\partial \mathbf{p}_{\alpha}}{\partial v_{i_l}}. \quad (3.42)$$

Le calcul du terme $\frac{\partial \mathbf{N}}{\partial v_{i_l}}$ correspond à l'évaluation de la dérivée partielle des fonctions de bases de l'entité B-spline par rapport à une composante non-triviale des vecteurs nodaux : l'algorithme, basé sur la méthode du *knot insertion* pour réaliser cette opération est disponible dans [139]. Dans [139], on peut également retrouver l'expression de $\frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_{\alpha}$ complète, correspondant à la dérivée partielle d'une entité B-spline par rapport à une composante non-triviale du vecteur nodal. Quant au terme $\frac{\partial \mathbf{p}_{\alpha}}{\partial v_{i_l}}$, il s'exprime à partir de l'Eq. (3.20) par :

$$\begin{aligned} \mathbf{N}\mathbf{p}_{\alpha} - \mathbf{q}_{\alpha} = 0 &\implies \frac{\partial \mathbf{N}\mathbf{p}_{\alpha}}{\partial v_{i_l}} = \frac{\partial \mathbf{q}_{\alpha}}{\partial v_{i_l}}, \\ &\iff \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_{\alpha} + \mathbf{N} \frac{\partial \mathbf{p}_{\alpha}}{\partial v_{i_l}} = 0, \\ &\iff \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_{\alpha} = -\mathbf{N} \frac{\partial \mathbf{p}_{\alpha}}{\partial v_{i_l}}, \\ &\iff \frac{\partial \mathbf{p}_{\alpha}}{\partial v_{i_l}} = -(\mathbf{N}^{\text{T}}\mathbf{N})^{-1} \mathbf{N}^{\text{T}} \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_{\alpha}. \end{aligned} \quad (3.43)$$

Ainsi, dans le cas de la formulation globale, l'Eq. (3.42) est reformulée et résolue directement par :

$$\frac{\partial \mathbf{h}_\alpha}{\partial v_{i_l}} = \left(\mathbf{I} - \mathbf{N} (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \right) \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_\alpha. \quad (3.44)$$

Dans le cas de la formulation par séparation de variables, le terme $\mathbf{N} \frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$ de l'Eq. (3.42) est calculé via deux analogies successives :

- la formulation matricielle d'une hypersurface B-spline est $\mathbf{h}_\alpha = \mathbf{N} \mathbf{P}_\alpha$. Ainsi, le calcul de $\mathbf{N} \frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$ est assimilé au calcul d'une hypersurface B-spline en remplaçant les coordonnées \mathbf{P}_α par $\frac{\partial \mathbf{P}_\alpha}{\partial v_{i_l}}$. Le passage de $\frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$ à $\frac{\partial \mathbf{P}_\alpha}{\partial v_{i_l}}$ se fait via l'Eq. (3.12) de l'indice linéaire ;
- la seconde analogie permet d'assimiler le terme $\frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_\alpha$ à \mathbf{Q}_α . Ainsi, le calcul de $(\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_\alpha$ de l'Eq. (3.43) est fait en considérant l'Eq. (3.24) et l'Eq. (3.12) de l'indice linéaire.

Comparaison des formulations dans l'évaluation du gradient de la fonction objectif par rapport aux valeurs non triviales des vecteurs nodaux

Pour évaluer l'impact des deux analogies nécessaires dans l'évaluation du gradient de la fonction objectif via la formulation par séparation des variables, l'exemple précédent est repris sans considérer de bruit artificiel. Ainsi, sa formulation est donnée par :

$$z = 2a \cos(\theta)^2 \sin(\phi). \quad (3.45)$$

Les paramètres des métamodèles sont également redéfinis et renseignés dans le Tab. 3.3. Les erreurs relatives entre les gradients évalués par les deux formulations sont renseignées dans le Tab. 3.4. On peut remarquer que cette différence ne dépasse pas 3 %. De plus, le temps total d'évaluation de ces neuf gradients est d'environ 30 min avec la formulation globale contre moins de 1 s avec la seconde formulation. Pour le reste de ce travail, la formulation par séparation de variables sera également considérée pour le calcul des gradients de la fonction objectif.

TABLE 3.3 – Caractéristiques communes aux métamodèles employés pour comparer les approches globale et par séparation des dimensions dans le calcul du gradient de la fonction objectif par rapport aux valeurs non triviales des vecteurs nodaux.

Variables	s_k	r_k	n_k
θ	3	12	8
ϕ	3	11	7

TABLE 3.4 – Erreur relative entre les gradients calculés par les deux formulations, pour l'étape de l'optimisation $l = 0$.

i	1	2	3	4	5	6	7	8	9
$\bar{\varepsilon}_{\text{sep, glob}} [\%]$	0.09	0.07	2.8	0.08	0.09	0.05	0.20	0.20	0.30

Résolution du CNLPP avec lissage

La résolution du CNLPP associé à ξ_2 , en prenant en compte le terme de lissage, revient à minimiser la valeur de la fonction objectif de l'Eq. (3.15) :

$$\frac{\partial f_{\text{obj, liss}}}{\partial \beta_{i_l}} = \sum_{\alpha=1}^M 2 (\mathbf{N} \mathbf{p}_\alpha - \mathbf{q}_\alpha)^T \frac{\partial \mathbf{h}_\alpha}{\partial v_{i_l}} (v_{i_{l-1}} - v_{i_{l+1}}) + \lambda \frac{\partial J_\alpha}{\partial \beta_{i_l}}, \quad (3.46)$$

Les deux termes complexes de cette expression sont $\frac{\partial \mathbf{h}_\alpha}{\partial v_{i_l}}$ et $\frac{\partial J_\alpha}{\partial \beta_{i_l}}$. La formulation du premier est décrite dans la section précédente à la différence du terme $\mathbf{N} \frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$ qui est recalculé à partir de l'Eq. (3.23) :

$$\begin{aligned} (\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E}) \mathbf{p}_\alpha = \mathbf{N}^T \mathbf{q}_\alpha &\Leftrightarrow \frac{\partial}{\partial v_{i_l}} \left((\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E}) \mathbf{p}_\alpha \right) = \frac{\partial}{\partial v_{i_l}} \left(\mathbf{N}^T \mathbf{q}_\alpha \right), \\ &\Leftrightarrow \frac{\partial}{\partial v_{i_l}} \left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right) \mathbf{p}_\alpha + \left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right) \frac{\partial \mathbf{p}_\alpha}{\partial v_{j_l}} = \frac{\partial \mathbf{N}^T}{\partial v_{j_l}} \mathbf{q}_\alpha, \\ &\Leftrightarrow \frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}} = \left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right)^{-1} (\mathbf{a} - \mathbf{b}), \end{aligned} \quad (3.47)$$

avec $\mathbf{a} = \frac{\partial \mathbf{N}^T}{\partial v_{i_l}} \mathbf{q}_\alpha$ et $\mathbf{b} = \frac{\partial}{\partial v_{i_l}} \left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right) \mathbf{p}_\alpha$. En considérant le formalisme de l'Eq. (3.27), le terme de droite de l'Eq. 3.47 devient dans la formulation par séparation de variables :

$$\left(\mathbf{N}^T \mathbf{N} + \lambda \mathbf{E} \right)^{-1} (\mathbf{a} - \mathbf{b}) \implies \left(\mathbf{N}_k^T \mathbf{N}_k + \lambda_k \mathbf{E}_k \right)^{-1} (\mathbf{a}_k - \mathbf{b}_k). \quad (3.48)$$

De nouveau, par analogie avec l'Eq. (3.27) et en considérant l'Eq. (3.12) des indices linéaires, il est possible de calculer le terme $\frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$ en remplaçant $\mathbf{N}_k^T \mathbf{Q}_\alpha^{k-1}$ par $(\mathbf{a}_k - \mathbf{b}_k)$ et \mathbf{P}_α par $\frac{\partial \mathbf{p}_\alpha}{\partial v_{i_l}}$. Le terme \mathbf{a}_k est calculé par $\mathbf{a}_k = \frac{\partial \mathbf{N}_k^T}{\partial v_{i_l}} \mathbf{Q}_\alpha^{k-1}$ par analogie avec $\frac{\partial \mathbf{N}}{\partial v_{i_l}} \mathbf{p}_\alpha$ et

l'algorithme du *knot insertion* de [139]. Quant au terme $\mathbf{b}_k = \frac{\partial}{\partial v_{i_l}} \left(\mathbf{N}_k^T \mathbf{N}_k + \lambda_k \mathbf{E}_k \right) \mathbf{P}_\alpha^{k-1}$, il s'exprime par :

$$\frac{\partial}{\partial v_{i_l}} \left(\mathbf{N}_k^T \mathbf{N}_k + \lambda_k \mathbf{E}_k \right) \mathbf{P}_\alpha^{k-1} = \left(\frac{\partial \mathbf{N}_k^T \mathbf{N}_k}{\partial v_{i_l}} + \lambda_k \frac{\partial \mathbf{E}_k}{\partial v_{i_l}} \right) \mathbf{P}_\alpha^{k-1}, \quad (3.49)$$

avec

$$\frac{\partial \mathbf{N}_k^T \mathbf{N}_k}{\partial v_{i_l}} = \frac{\partial \mathbf{N}_k^T}{\partial v_{i_l}} \mathbf{N}_k + \mathbf{N}_k^T \frac{\partial \mathbf{N}_k}{\partial v_{i_l}} = 2 \mathbf{N}_k^T \frac{\partial \mathbf{N}_k}{\partial v_{i_l}}, \quad (3.50)$$

et

$$\frac{\partial \mathbf{E}_k}{\partial v_{i_l}} = \int_0^1 2 \frac{\partial}{\partial v_{i_l}} \left(\frac{\partial^2 \mathbf{N}_k}{\partial \zeta_k^2} \right) \frac{\partial^2 \mathbf{N}_k}{\partial \zeta_k^2} d\zeta_k. \quad (3.51)$$

La dérivée seconde par rapport au paramètre adimensionné ζ_k des polynômes de Bernstein regroupés dans la matrice \mathbf{N} se retrouve dans [24, 138]. Cette dérivée correspond à celle d'une courbe dans la mesure où on considère la formulation par séparation des dimensions.

Pour finir, le gradient du terme de lissage par rapport à un coefficient β_{i_l} , en considérant la formulation par séparation de dimensions, s'exprime par :

$$\frac{\partial J_\alpha}{\partial \beta_{i_l}} = \sum_{k=1}^N \frac{\partial J_\alpha}{\partial \|\mathbf{J}_{\alpha,k}\|_s} \frac{\partial \|\mathbf{J}_{\alpha,k}\|_s}{\partial \beta_{i_l}} \quad (3.52)$$

avec $\frac{\partial J_\alpha}{\partial \|\mathbf{J}_{\alpha,k}\|_s} = 1$ et $\frac{\partial \|\mathbf{J}_{\alpha,k}\|_s}{\partial \beta_{i_l}}$ exprimé par :

$$\frac{\partial \|\mathbf{J}_{\alpha,k}\|_s}{\partial \beta_{i_l}} = (v_{i_l-1} - v_{i_l+1}) (\|\mathbf{J}_{\alpha,k}\|_s)^{1-s} \sum_{\kappa_{\text{CP},1}^*=0}^{n_{\text{CP}}^*} \sum_{\kappa_{\text{CP},2}^*=0}^{n_{\text{CP}}^*} J_{\alpha,k,\kappa_{\text{CP},1}^*,\kappa_{\text{CP},2}^*}^{s-1} \frac{\partial J_{\alpha,k,\kappa_{\text{CP},1}^*,\kappa_{\text{CP},2}^*}}{\partial v_{i_l}}, \quad (3.53)$$

où $n_{\text{CP}}^* = \prod_{\substack{k^*=1 \\ k^* \neq k}}^N (n_{k^*} + 1)$. Le terme $\frac{\partial J_{\alpha,k,\kappa_{\text{CP},1}^*,\kappa_{\text{CP},2}^*}}{\partial v_{i_l}}$ dérive directement de $\frac{\partial \mathbf{J}_{\alpha,k}}{\partial \beta_{i_l}}$ de l'Eq. (3.31) :

$$\begin{aligned} \frac{\partial \mathbf{J}_{\alpha,k}}{\partial v_{i_l}} &= \frac{\partial \mathbf{P}_{\alpha,k}^{*\text{T}} \mathbf{E}_k \mathbf{P}_{\alpha,k}^*}{\partial v_{i_l}} \\ &= 2 \mathbf{P}_{\alpha,k}^{*\text{T}} \mathbf{E}_k \frac{\partial \mathbf{P}_{\alpha,k}^*}{\partial v_{i_l}} + \mathbf{P}_{\alpha,k}^{*\text{T}} \frac{\partial \mathbf{E}_k}{\partial v_{i_l}} \mathbf{P}_{\alpha,k}^*, \end{aligned} \quad (3.54)$$

avec $\frac{\partial \mathbf{E}_k}{\partial v_{i_l}}$ exprimé par l'Eq. (3.51).

La mise en place de cette procédure d'optimisation séquentielle est représentée schématiquement dans la Fig. 3.5. Les coordonnées des CPs étant dépendantes des composantes des vecteurs nodaux \mathbf{v}_k d'après l'Eq. (3.24), elles sont recalculées à chaque itération de l'algorithme au gradient. Les étapes de ce diagramme de flux sont décrites ci-dessous :

1. évaluation des vecteurs nodaux initiaux comme entrée de l'algorithme d'optimisation et, par conséquent, initialisation de ξ_2 . Ces vecteurs sont initialisés uniformes ou

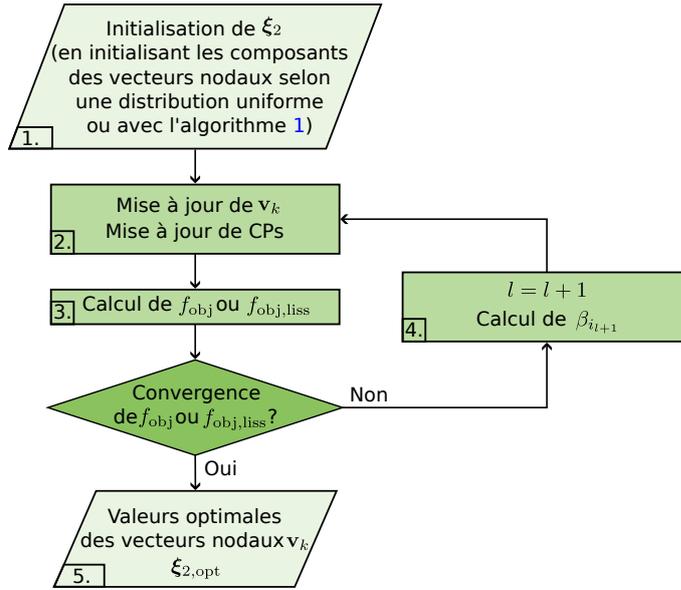


FIGURE 3.5 – Diagramme de flux de la procédure d’optimisation de la fonction objectif par rapport aux composantes non-triviales des vecteurs nodaux.

coïncidants avec la répartition de la base de données via l’algorithme 1 ;

2. calcul des valeurs non-triviales des vecteurs nodaux à partir des coefficients β_{i_l} et de l’Eq. (3.37). Calcul des coordonnées des CPs en fonction des valeurs mises à jour des vecteur nodaux et de l’Eq. (3.24) ou de l’Eq. (3.27) si le lissage est pris en compte ;
3. calcul de f_{obj} ou de $f_{obj,liss}$ si le terme de lissage est pris en compte ;
4. si la valeur de f_{obj} , ou de $f_{obj,liss}$, n’a pas convergé vers un minimum local, mise à jour des coefficients $\beta_{i_{l+1}}$;
5. sortie de l’algorithme d’optimisation avec les valeurs non-triviales des vecteurs nodaux optimisées ainsi que les coordonnées des CPs associées.

3.4.2.2 Optimisation des poids

La dernière phase d’optimisation de la Fig. 3.1 correspond à l’optimisation des poids associés aux coordonnées des CPs. Cette optimisation diffère des précédentes principalement parce qu’elle se base sur une hypersurface NURBS. Qualitativement, il s’agit de modifier l’importance de certains CPs en manipulant l’impact des polynômes de Bernstein du support local associé, défini par l’Eq. (3.7). Les variables à optimiser sont contenues dans le vecteur :

$$\xi_3 = \{\omega_{0,\dots,0}, \dots, \omega_{i_1,\dots,i_N}, \dots, \omega_{n_1,\dots,n_N}\}, \xi_3 \in \mathbb{R}^{n_{CP}}, \quad (3.55)$$

dont les valeurs sont initialisées comme étant toutes égales ou selon des règles empiriques présentées en partie dans [9] et énoncées dans [178]. En outre, ces poids sont contraints

par l'inégalité $\omega_{i_1, \dots, i_N} > 0$, considérée indirectement dans la définition de leurs bornes de variations. Ces dernières sont fixées à $\omega_{\text{LB}} = 1$ pour la borne inférieure et à $\omega_{\text{UB}} = 10$ pour la borne supérieure.

La résolution du problème CNLPP associé à ξ_3 revient également à minimiser la valeur de la fonction objectif de l'Eq. (3.13) par :

$$\min_{\xi_3} (f_{\text{obj}}) \Rightarrow \frac{\partial f_{\text{obj}}}{\partial \xi_3} = 0. \quad (3.56)$$

La résolution de cette équation ne comportant pas de solution analytique explicite, l'emploi d'un algorithme au gradient est nécessaire. Il permet d'approcher la solution du problème de l'Eq. (3.56) par itérations successives via le calcul du gradient de f_{obj} dans la direction de sortie α :

$$\begin{aligned} \frac{\partial f_{\text{obj}, \alpha}}{\partial \omega_{i_1, \dots, i_N}} &= \left(\frac{\partial}{\partial \omega_{i_1, \dots, i_N}} (\mathbf{Np}_\alpha - \mathbf{q}_\alpha)^T (\mathbf{Np}_\alpha - \mathbf{q}_\alpha) \right) \\ &= 2 (\mathbf{Np}_\alpha - \mathbf{q}_\alpha)^T \frac{\partial \mathbf{h}_\alpha}{\partial \omega_{i_1, \dots, i_N}} \end{aligned} \quad (3.57)$$

avec

$$\frac{\partial \mathbf{h}_{\alpha, j}}{\partial \omega_{i_1, \dots, i_N}^{(l)}} = \frac{N_{i_1, s_1}(\zeta_{1, j}) \times \dots \times N_{i_N, s_N}(\zeta_{N, j})}{W} \left[p_{i_1, \dots, i_N}^\alpha \text{diag}(\mathbf{I}_{n_{\text{TP}}}) \mathbf{1} - \mathbf{Np}_\alpha \right] \quad (3.58)$$

où $\mathbf{I}_{n_{\text{TP}}} \in \mathbb{R}^{n_{\text{TP}} \times n_{\text{TP}}}$ est la matrice identité de taille n_{TP} et sa diagonale représente un vecteur de longueur n_{TP} dont les valeurs sont égales à l'unité et

$$W = \sum_{j_1=0}^{n_1} \dots \sum_{j_N=0}^{n_N} N_{j_1, s_1}(\zeta_{1, j}) \times \dots \times N_{j_N, s_N}(\zeta_{N, j}) \times \omega_{j_1, \dots, j_N}. \quad (3.59)$$

La mise en place de cette optimisation suit le diagramme de flux de la Fig. 3.6 dont les étapes sont énumérées ci-dessous :

1. initialisation de l'algorithme par les valeurs des coordonnées des CPs et des vecteurs nodaux issus des étapes d'optimisation précédentes ;
2. initialisation des poids à une même valeur commune ;
3. évaluation de la fonction objectif f_{obj} ;
4. si la fonction objectif n'a pas convergé vers un minimum local, calcul des nouvelles valeurs de $\omega_{i_1, \dots, i_N}^{(l+1)}$ en fonction des valeurs de $\omega_{i_1, \dots, i_N}^{(l)}$ et du gradient de la fonction objectif par rapport à $\omega_{i_1, \dots, i_N}^{(l)}$ via l'Eq. (3.57) et l'algorithme GC-MMA ;
5. sortie de l'algorithme d'optimisation avec les valeurs des poids ω associées à chaque CP optimisés, contenus dans le vecteur ξ_3 .

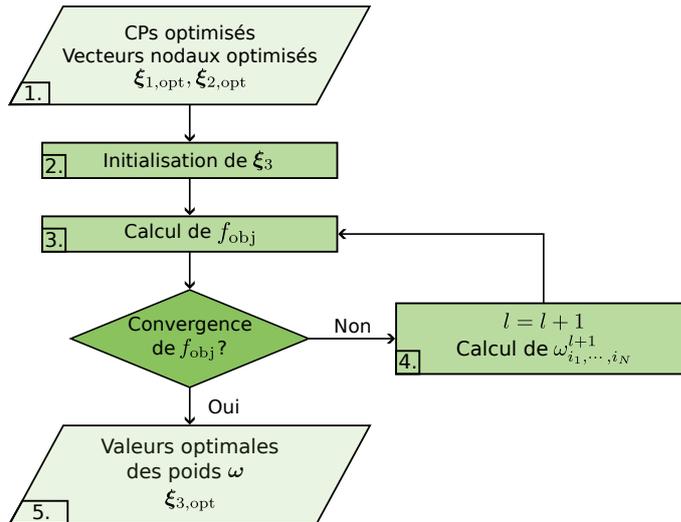


FIGURE 3.6 – Diagramme de flux du fonctionnement de l’algorithme d’optimisation du métamodèle par rapport aux poids.

Remarque 2 Les coûts de calcul de chacune des étapes d’optimisation vont croissants. L’optimisation de ξ_1 nécessite quelques secondes (1–5 s) là où celle de ξ_2 quelques minutes (10–15 min) et celle de ξ_3 quelques heures (2–3 h).

3.5 Application du métamodèle sur des cas analytiques représentatifs

Cette section propose d’illustrer le rôle de chaque étape du processus d’optimisation présenté dans ce chapitre, sur des cas analytiques représentatifs. Le premier cas test est une surface fermée générée par l’équation paramétrique d’un cylindre. L’objectif est d’illustrer le rôle du terme de lissage introduit lors du calcul des CPs. Le second cas test est une fonction créneau qui illustre l’influence de l’optimisation des valeurs non-triviales des vecteurs nodaux. Cette fonction est très non-linéaire et se retrouve dans bon nombre de problèmes physiques. Le dernier cas test correspond à la fonction mathématique de Salomon, reproduisant la propagation d’une onde acoustique à la surface d’un fluide. Son expression est tirée de [150] et est fortement non-linéaire. Ce dernier cas test met en évidence le rôle de l’optimisation des poids de l’entité NURBS associés aux coordonnées des CPs.

En outre, il est important de faire la distinction entre l’emploi des entités NURBS dans un contexte de métamodélisation et celui d’un problème de *fitting*. Dans le premier cas, le but est de reproduire le comportement du modèle initial en utilisant le moins de ressources de calcul possible (base de données, temps de calcul, etc.). Une base d’entraînement est employée pour calibrer le métamodèle et une de contrôle pour mesurer sa précision. Dans le second cas, l’objectif est que l’hypersurface s’approche au plus juste de l’ensemble des données disponibles, dans la base d’entraînement. Un phénomène d’*overfitting* peut alors apparaître : les données de la base d’entraînement sont parfaitement retrouvées mais le

TABLE 3.5 – Caractéristiques des métamodèles employés pour reproduire la forme du cylindre bruitée.

Variable	s_k	r_k	n_k
z	3	20	16
θ	3	50	40

comportement de l’hypersurface en dehors de ces points est erroné. Tout le travail consiste alors à calibrer au mieux les variables entières du métamodèle et à utiliser intelligemment et efficacement le processus d’optimisation, tout en évitant l’*overfitting*.

Remarque 3 *Les valeurs des fonctions objectifs fournies dans la suite sont dé-normées par rapport à leurs expressions initiales.*

3.5.1 Influence du lissage sur les points de contrôle

Un cylindre est une surface fermée, évoluant dans un espace euclidien. Ce modèle mathématique MIMO est caractérisé par $N = 2$ variables d’entrée, θ et z , et $M = 3$ variables de sortie, x , y et z . Du bruit est introduit artificiellement sur les sortie x et y pour mettre en évidence l’efficacité du terme de lissage. La formulation du cylindre est alors donnée par :

$$\begin{cases} x = R \cos(\theta) + \Phi, \\ y = R \sin(\theta) + \Phi, \text{ avec } \theta \in [0, 2\pi], \\ z = z, \end{cases} \quad (3.60)$$

avec Φ une variable aléatoire évoluant selon une loi uniforme centrée en 0 définie sur $\mathcal{U}[-0.05; 0.05]$. Deux métamodèles, avec et sans terme de lissage, sont générés en considérant seulement la première étape du processus, et illustrés respectivement par les Figs. 3.7c et 3.7d. Leurs caractéristiques communes sont regroupées dans le Tab. 3.5. Pour déduire le nombre de CPs par rapport aux TPs, le ratio $\tau = \frac{2}{3}$ est employé [38]. Quant aux valeurs non-triviales des vecteurs nodaux, dans la mesure où les variables d’entrée sont connues et réparties uniformément sur leur ensemble de variation respectif, ils sont initialisés comme uniformes.

La Fig. 3.7a représente la surface initiale contenue dans la base de données \mathbf{Q}_t et la Fig. 3.7b son équivalent bruitée. La particularité de ce dernier est que les TPs pour $\theta = 0$ sont définis égaux aux TPs pour $\theta = 2\pi$: tout en ayant du bruit numérique, la surface reste une surface fermée.

La Fig. 3.8 trace la totalité des valeurs des TPs de la base de données en fonction de la totalité des valeurs calculées aux mêmes coordonnées par les métamodèles, en rouge avec le terme de lissage, sinon en vert. La droite $y = x$ est tracée en noir comme référence. On remarque que les valeurs générées par le second modèle sont plus proches de la courbe de référence que celles évaluées par le premier.

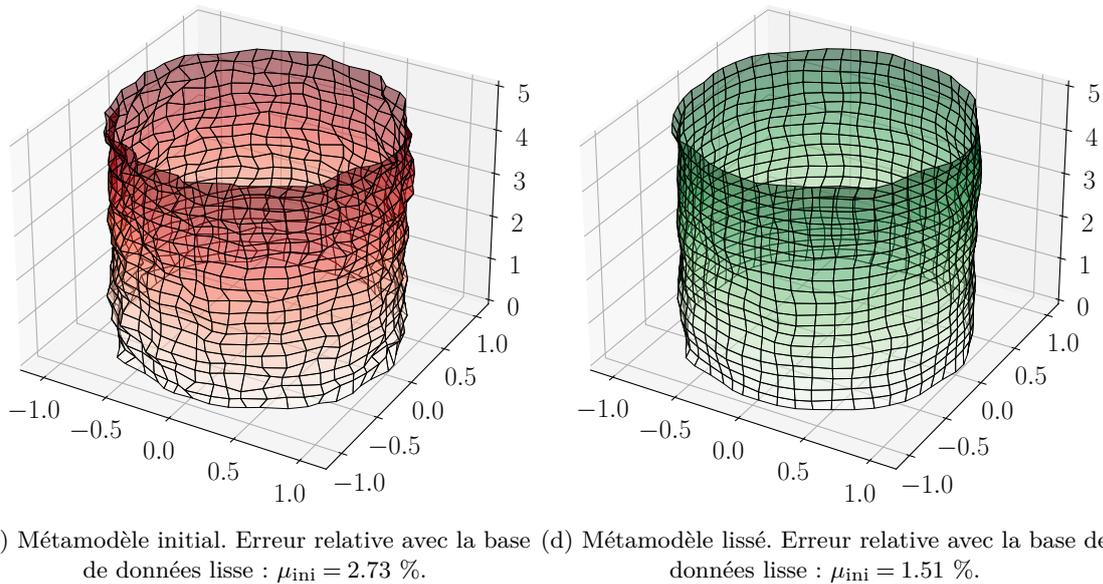
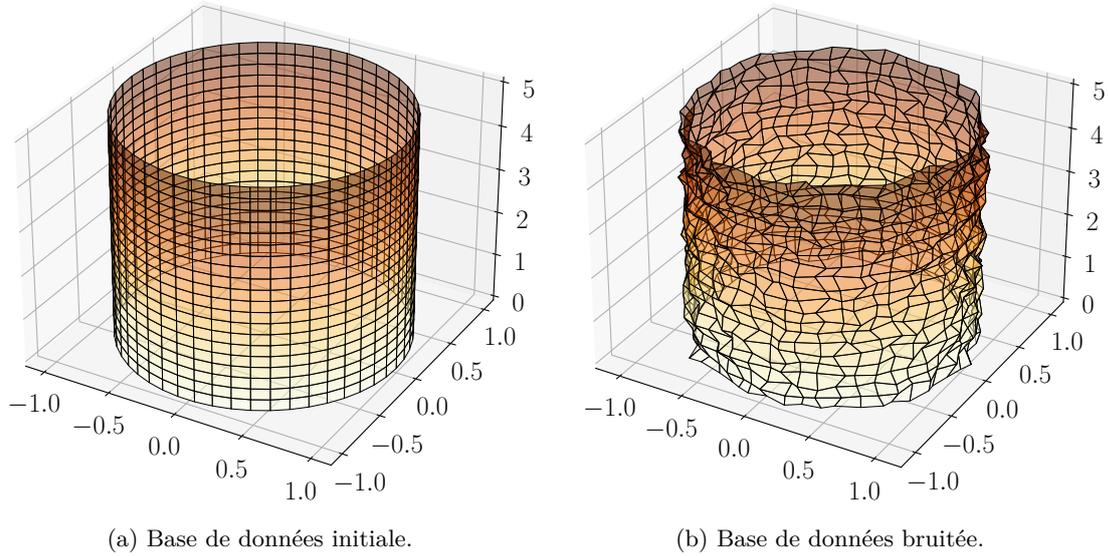


FIGURE 3.7 – Illustration de l’effet du terme de lissage sur le comportement du métamodèle. En (a) la surface initiale, en (b) la surface comportant du bruit parasite, en (c) la surface générée par le métamodèle non lissé et en (d) la surface générée par le métamodèle incluant le terme de lissage.

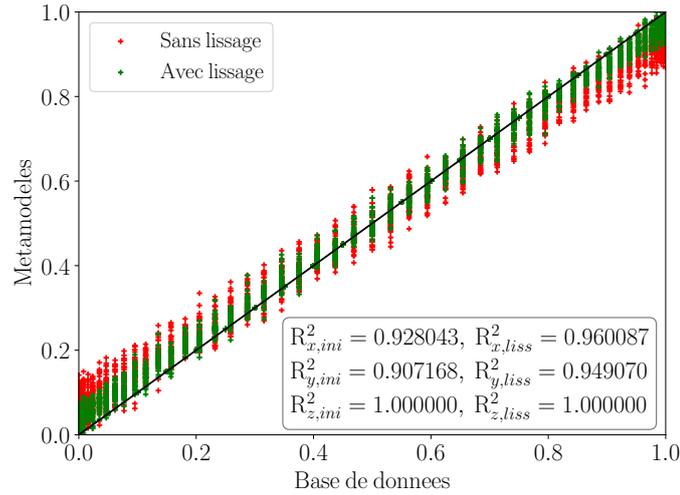


FIGURE 3.8 – Graphique représentant les *Target Points* de la base de données lisse en fonction des *Target Points* évalués par le métamodèle initial en rouge et lissé en vert.

D'un point de vue quantitatif, les coefficients R^2 du second métamodèle évalués par rapport à $\mathbf{Q}_{t,liss}$ sont plus élevés en moyenne de 4.00 % par rapport à ceux du premier métamodèle. Quant aux erreurs relatives moyennes, elles sont de $\bar{\varepsilon} = 2.73$ % pour le cas sans lissage et de $\bar{\varepsilon}_{liss} = 1.51$ % pour le cas lissé.

3.5.2 Influence de l'optimisation des vecteurs nodaux et *overfitting*

3.5.2.1 Optimisation des vecteurs nodaux

La modélisation de certains phénomènes physiques requiert de reproduire le comportement non-linéaire d'une fonction créneau. On peut retrouver cette situation dans l'étude de la propagation d'une onde acoustique dans le cas de référence du tube à choc de Sod [166]. On retrouve également ce type de discontinuités lors de la propagation d'un front de changement de phase lors d'un changement d'état [4]. Mathématiquement, ce type de non-linéarité peut s'exprimer par :

$$\begin{cases} y(x) = \bar{y} \text{ si } x(t) \in [x_{LB}, x_{UB}], \\ y(x) = 0 \text{ sinon.} \end{cases} \quad (3.61)$$

Pour l'exemple illustré dans la suite, $\bar{y} = 1$, $x_{LB} = 9$, $x_{UB} = 18$. Ce modèle mathématique *Single-Input-Multiple-Output* (SIMO) est caractérisé par $N = 1$ variable d'entrée, et $M = 2$ variables de sortie. Les caractéristiques du métamodèle employé sont regroupées dans le Tab. 3.6, ainsi que les bornes de variations des variables d'optimisation β_i . Le degré de p_k est fixé à $p_k = 3$ et le ratio τ est fixé à $\tau = \tau_k = \frac{2}{3}$ [38]. Quant aux valeurs non-triviales du vecteur nodal, dans la mesure où les valeurs de la variable d'entrée t ne sont pas connues, elles sont initialisées via l'algorithme 1 et la méthode *Centripetal* tirée de [108].

Le résultat de l'optimisation des valeurs non-triviales du vecteur nodal sont représen-

TABLE 3.6 – Caractéristiques du métamodèle employé pour reproduire la forme de la fonction créneau.

Variable	s_k	r_k	n_k	β_i
t	3	31	20	$]0, 1[$

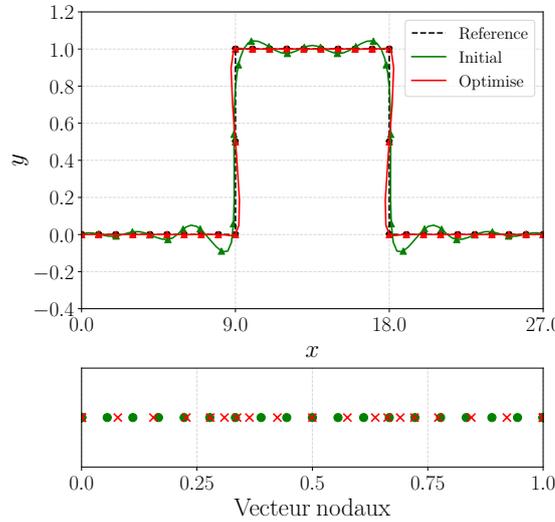


FIGURE 3.9 – Métamodèle appliqué à une fonction créneau. La référence de cette fonction est tracée en pointillés noirs avec les points de la base d’entraînement \mathbf{Q}_t repérés. En vert, il s’agit du métamodèle avant optimisation des valeurs non triviales du vecteur nodal, en rouge après cette optimisation. On peut voir également sur les figures du bas, les vecteurs nodaux avant et après optimisation.

tées dans la Fig. 3.9. On peut y voir la fonction initiale en pointillés noirs avec les TPs employés pour l’entraînement du métamodèle. En vert, le métamodèle avant optimisation des composantes du vecteur nodal et rouge après optimisation. Visuellement, les résultats sont concluants : l’optimisation permet au métamodèle de se rapprocher des TPs de la base de données d’entraînement, repérés par les marqueurs. La répartition des valeurs non-triviales des vecteurs nodaux avant et après l’optimisation est également représentée en dessous. Les évolutions des indicateurs sont données par le Tab. 3.7. On peut constater que les coefficients R^2 tendent vers 1 après optimisation et les erreurs relatives moyennes vers 0.

TABLE 3.7 – Évolution des coefficients f_{obj} , R^2 et $\bar{\varepsilon}$ du métamodèle reproduisant la forme de la fonction créneau.

Variable	f_{obj}		R^2		$\bar{\varepsilon}$ [%]	
	ξ_1	ξ_2	ξ_1	ξ_2	ξ_1	ξ_2
x	0.2611	1.67×10^{-7}	0.9999	1.000	0.5983	1.92×10^{-3}
y			0.9359	0.9999	4.2188	8.3708×10^{-3}

En considérant ces observations, on peut conclure que le meilleur moyen d’approximer

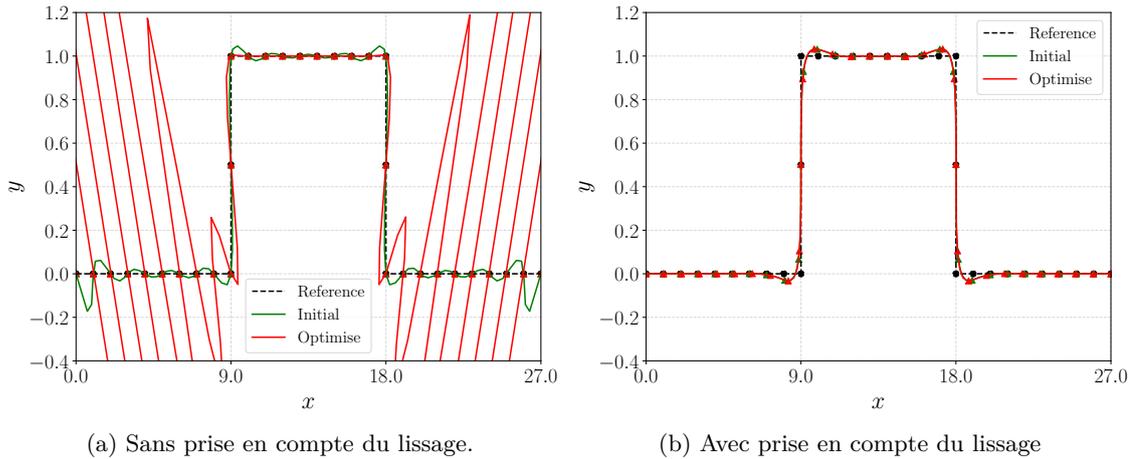


FIGURE 3.10 – Métamodèle appliqué à une fonction créneau. La référence de cette fonction est tracée en pointillés noirs avec les points de la base d’entraînement \mathbf{Q}_t repérés. En vert, il s’agit du métamodèle avant optimisation des valeurs non triviales du vecteur nodal, en rouge après cette optimisation.

cette fonction serait d’utiliser une entité B-spline de degré $p_k = 1$ avec six CPs dont le calcul se fait « manuellement » : $\mathbf{P} : [(x_{\min}, 0), (x_{LB}, 0), (x_{LB}, \bar{y}), (x_{UB}, \bar{y}), (x_{UB}, 0), (x_{\max}, 0)]$.

3.5.2.2 Réduction du phénomène d’*overfitting*

En considérant le même problème que pour la section précédente, deux nouveaux métamodèles sont construits, un en considérant le terme de lissage J_α , l’autre non. Leurs caractéristiques sont identiques au cas précédent (cf. Tab. 3.6) mais le nombre de CPs est donnée par : $n_k = r_k - 1$. Le but est d’illustrer le rôle du terme de lissage dans la réduction du phénomène d’*overfitting*.

La Fig. 3.10 montre les résultats fournis par les deux métamodèles et le Tab. 3.8 rassemble les résultats numériques. On peut voir sur la Fig. 3.10a une situation d’*overfitting* manifeste, bien que les indicateurs d’erreur entre le métamodèle sans terme de lissage et sa base d’entraînement soient quasiment parfaits : $R^2 = 1.0000$. Quant à la Fig. 3.10b elle illustre l’impact du terme de lissage sur le comportement du métamodèle : les fortes oscillations présentes dans la première situation sont complètement absorbées bien que les valeurs des indicateurs soient moins bonnes. Dans des cas similaires, il sera alors préférable d’avoir une moins bonne approximation de la base de données d’entraînement plutôt qu’être en présence d’un phénomène d’*overfitting*.

3.5.3 Influence de l’optimisation du poids

La fonction de propagation d’onde acoustique à la surface d’un fluide, ou fonction *Drop-Wave* en anglais, aussi connue sous le nom de fonction de Salomon [150], est une

TABLE 3.8 – Évolution des coefficients f_{obj} , R^2 et $\bar{\varepsilon}$ pour les métamodèles avec et sans terme de lissage.

	Variable	f_{obj}		R_α^2		$\bar{\varepsilon}_\alpha$ [%]	
		ξ_1	ξ_2	ξ_1	ξ_2	ξ_1	ξ_2
Sans lissage	x	4.95×10^{-4}	3.54×10^{-7}	0.9999	1.000	0.0150	0.0001
	y			1.000	1.000	0.0000	0.0000
Avec lissage	x	0.35	0.27	0.9999	0.9996	0.2305	1.3134
	y			0.9711	0.9415	1.8708	2.5580

fonction fortement non-linéaire, en deux dimensions, donnée par :

$$u(x_1, x_2) = 1 - \cos\left(2\pi\sqrt{x_1^2 + x_2^2}\right) + 0.1\sqrt{x_1^2 + x_2^2}, \quad (3.62)$$

avec $(x_1, x_2) \in \mathcal{U}[-10, 10]$ les coordonnées de l'espace des entrées. Le problème est caractérisé par $N = 2$ variables d'entrée et $M = 1$ variable de sortie. Le nombre total de TPs est $n_{\text{TP}} = (r_1 + 1)(r_2 + 1) = 2916$, et le degré du polynôme de Bernstein est fixé à $s_k = 2$ pour chacune des deux directions d'entrée $k = 1, 2$. Pour souligner l'importance du nombre de CPs sur la précision du résultat, deux métamodèles sont évalués : le premier avec un ratio τ égal à $\tau = \frac{2}{3}$ [38] et un second avec un rapport $\tau = \frac{4}{5}$. Les variables de conception du métamodèle ainsi que les bornes de variation des variables d'optimisation β_i et ω_{i_1, \dots, i_N} , sont renseignées dans le Tab. 3.9.

TABLE 3.9 – Caractéristiques des métamodèles employés pour la fonction *Drop-Wave*.

	Variable	s_k	r_k	n_k	β_i	ω_{i_1, \dots, i_N}
M 1	(x_1, x_2)	2	53	43	$]0, 1[$	$[1, 10]$
M 2	(x_1, x_2)	2	53	47	$]0, 1[$	$[1, 10]$

Le Tab. 3.10 contient les valeurs des différents indicateurs de précision du métamodèle. La Fig. 3.11 représente les écarts $\mathbf{e}_{M, \alpha}$ de l'Eq. (3.36), entre les différents métamodèles à chaque étape de leur optimisation respective et la base \mathbf{Q}_t . Pour finir, la Fig. 3.12, trace les valeurs calculées par le métamodèle en fonction des données de la base de données en considérant les trois phases de l'optimisation respectivement en bleu, vert et rouge.

TABLE 3.10 – Évolution des coefficients f_{obj} , R^2 et $\bar{\varepsilon}$, en fonction de l'étape d'optimisation et du nombre de CPs considéré.

M	f_{obj}			R_1^2			$\bar{\varepsilon}_1$ [%]		
	ξ_1	ξ_2	ξ_3	ξ_1	ξ_2	ξ_3	ξ_1	ξ_2	ξ_3
1	181.28	117.54	76.60	0.8799	0.9253	0.9525	16.06	11.34	9.27
2	82.38	16.55	4.99	0.9488	0.9901	0.9970	8.62	4.07	2.45

Les Fig. 3.11 et Fig. 3.12 montrent que les métamodèles deviennent de plus en plus

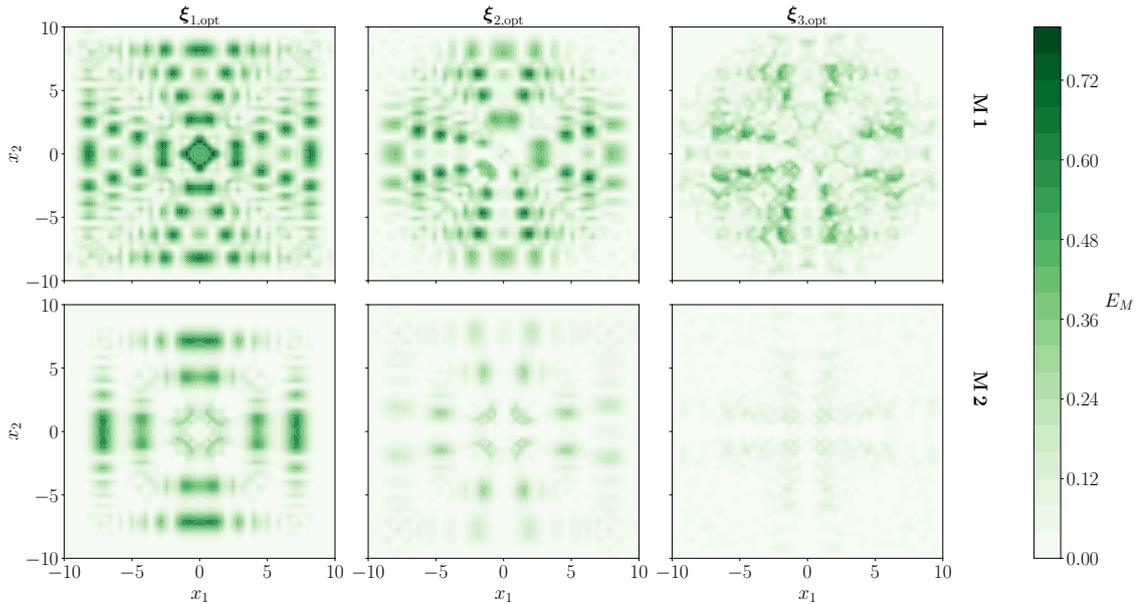


FIGURE 3.11 – Évolution des écarts entre les données fournies par les métamodèles et celles de la base de données d’entraînement en fonction des étapes d’optimisation.

précis au fur et à mesure que leurs variables de conception sont optimisées par rapport à la base d’entraînement \mathbf{Q}_t . En outre, les résultats sont meilleurs pour une valeur de τ plus élevée. Ce cas test met en évidence le rôle et l’efficacité de la stratégie d’optimisation. Néanmoins, nous verrons dans le chapitre suivant que lorsque le métamodèle est utilisé en tant que tel et non pas dans un environnement de *fitting*, i.e., avec un contrôle de sa précision via une base de contrôle \mathbf{Q}_c différente de celle d’entraînement \mathbf{Q}_t , les observations seront contradictoires.

3.6 Conclusion

Ce chapitre a permis d’expliciter la procédure de génération du métamodèle basé sur des hypersurfaces NURBS comme solution d’un problème d’optimisation. Après avoir rappelé la formulation générale des entités NURBS puis du cas particulier des B-splines, la stratégie d’optimisation en trois étapes a été définie. Elle vise à optimiser les variables de conception continues des entités NURBS pour obtenir un métamodèle précis tout en restant raisonnable sur la quantité de données nécessaire à son fonctionnement. Le métamodèle a ensuite été appliqué sur des cas analytiques à titre d’illustration.

Les avantages apportés par ces optimisations successives sont nombreux. La détermination des coordonnées des CPs par résolution du CNLPP associé permet de respecter la densité de répartition des TPs. De plus, l’ajout du terme de lissage permet de réduire l’effet de non-linéarités parasites de la base de données sur le métamodèle ainsi que le phénomène d’*overfitting*. L’optimisation des vecteurs nodaux permet de pleinement tirer avantage de la propriété de support local des hypersurfaces NURBS. Comme on peut le

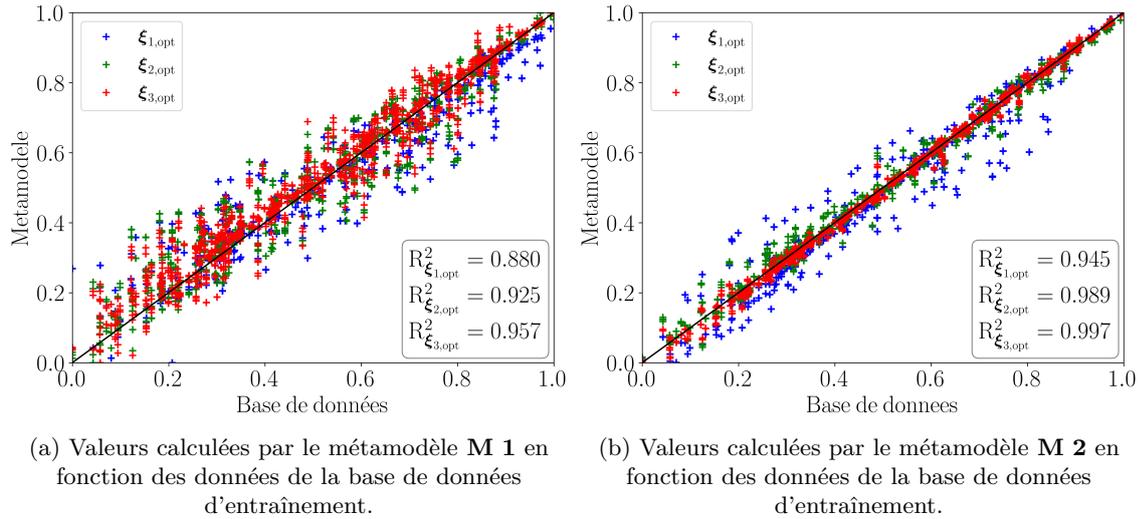


FIGURE 3.12 – Évolution de la précision des métamodèles en fonction de leur étape d'optimisation.

voir sur le cas test de la fonction créneau dans la section 3.5.2, les valeurs non-triviales des vecteurs nodaux vont être réorganisées pour apporter de la maniabilité à l'entité NURBS dans les zones de non-linéarités.

Pour finir, l'optimisation des poids permet d'intervenir sur l'hypersurface sans modifier les autres grandeurs déjà optimisées. Ainsi, une fois les CPs fixés, leur poids est modifié pour leur donner plus ou moins d'importance. Les résolutions approchées numériques des deux derniers CNLPP sont mises en place via l'algorithme au gradient employant la méthode GC-MMA. Cette méthode est robuste et permet notamment de considérer un grand nombre de variables d'optimisation.

Grâce à la mise en place de cette procédure d'optimisation, l'utilisateur est abjecte de calibration des variables continues du métamodèle. Ainsi, il peut se concentrer sur la définition du problème à traiter et appliquer la bonne calibration au métamodèle de manière objective. Le chapitre 4 qui suit, propose l'application de ce métamodèle sur différentes configurations, allant d'un modèle relativement linéaire à un problème fortement non-linéaire et d'un problème à faibles dimensions à un problème à grandes dimensions. Les avantages et inconvénients de cette stratégie seront alors mis en évidence, notamment par le biais du contrôle de la précision du métamodèle sur une base de données \mathbf{Q}_c , et par la comparaison des temps de simulation du modèle initial et du métamodèle.

4

Applications de la stratégie de métamodélisation basée sur les entités NURBS

Sommaire

4.1	Introduction	123
4.2	Cas analytiques physiques non-linéaires	124
4.2.1	Équation de Poisson	124
4.2.1.1	Description du problème	124
4.2.1.2	Caractéristiques et solution du métamodèle	125
4.2.2	Système oscillant non-linéaire	127
4.2.2.1	Description du problème	127
4.2.2.2	Caractéristiques et solution du métamodèle	128
4.2.3	Système oscillant non-linéaire : évolution temporelle	129
4.2.3.1	Description du problème	129
4.2.3.2	Caractéristiques et solution du métamodèle	130
4.2.4	Propagation d'une onde à la surface d'un fluide	132
4.2.4.1	Description du problème	132
4.2.4.2	Caractéristiques et solution du métamodèle	133
4.3	Problèmes mécaniques nécessitant une résolution numérique	136
4.3.1	Plaque pleine	136
4.3.1.1	Description du problème	136
4.3.1.2	Caractéristiques et solution du métamodèle	136
4.3.2	Plaque percée	138
4.3.2.1	Description du problème	138
4.3.2.2	Caractéristiques et solution du métamodèle	139
4.3.3	Panneau raidi en flambement	142
4.3.3.1	Description du problème	142
4.3.3.2	Caractéristiques et solution du métamodèle	144
4.4	Conclusion du chapitre	148

RÉSUMÉ DU CHAPITRE

Ce chapitre propose de mettre en application la méthode de génération d'un métamodèle basé sur les hypersurfaces Non-Uniform Rational Basis-Spline (NURBS). La première partie traite des cas physiques analytiques faiblement non-linéaires à fortement non-linéaires. Le but est de démontrer les capacités de la méthode à la résolution d'une grande variété de problèmes en mécanique et en physique, en présence de singularités, soit géométriques, soit liées à l'inclusion de phénomènes non-linéaires. Le métamodèle est ensuite appliqué à des problèmes de mécanique simulés à l'aide de la méthode aux Éléments Finitis (EF). L'objectif est de montrer les capacités du métamodèle à traiter des systèmes de grande dimension et à grand nombre de degrés de liberté tout en restituant correctement les singularités locales. Enfin, le niveau de précision et les temps de simulation des modèles haute-fidélités et des métamodèles associés seront comparés.

4.1 Introduction

La procédure de génération d'un métamodèle basé sur les hypersurfaces NURBS comme solution d'un problème d'optimisation sous contraintes a été présentée au chapitre 3. Pour rappel, la procédure se déroule en trois étapes :

- optimisation des coordonnées des CPs, stockées dans un vecteur ξ_1 ;
- optimisation des valeurs non triviales des vecteurs nodaux, stockées dans un vecteur ξ_2 ;
- optimisation des poids associés aux CPs, stockés dans un vecteur ξ_3 .

Ce chapitre propose à présent d'utiliser la méthode pour illustrer le large spectre possible d'applications. Le but est de mettre en évidence la possibilité de retrouver les détails d'un modèle avec une très bonne finesse de résolution ainsi que sa rapidité d'emploi.

Dans la première partie, section 4.2, les applications admettent des solutions analytiques, y compris en présence de singularités. L'objectif est de montrer la polyvalence du métamodèle à traiter des non-linéarités. Le premier cas étudié correspond à la solution de l'équation de Poisson en deux dimensions [34], qui est un cas faiblement non-linéaire. L'étude suivante porte sur un système non-linéaire oscillant : le mouvement temporel d'une masse sans glissement, reliée à un support par deux ressorts et un amortisseur [23,132,147]. Une première version du problème consiste à reproduire les amplitudes maximales atteintes par le modèle initial via le métamodèle. La seconde version consiste à reproduire le comportement temporel de l'oscillateur via le métamodèle. Le dernier cas analytique traité correspond à l'équation de Salomon [150] employée au chapitre 3 pour illustrer le rôle de chaque étape du processus d'optimisation générant le métamodèle. Cette fonction représente la propagation d'une onde de choc à la surface d'un fluide : son comportement est très fortement non-linéaire.

Dans la seconde partie, les applications visées ont un grand nombre de degrés de liberté. Les données d'entrée de ces problèmes sont numériques, obtenues avec une résolution numérique en passant par la méthode aux EF. Ces modèles étant coûteux à mettre en place, la quantité de données disponibles est faible. Le premier correspond à l'étude du champ de déplacement d'une plaque pleine en flexion en prenant en compte les non-linéarités géométriques et du matériau. Il s'agit d'un cas faiblement non-linéaire. Le second problème est l'étude du champ de déplacement d'une plaque percée en flexion. Il s'agit de la même situation que le cas précédent mais avec une singularité géométrique pouvant être traitée de différentes façons. Le dernier cas d'étude propose d'appliquer la méthode de métamodélisation sur le modèle numérique haute-fidélité EF d'un panneau raidi soumis à un chargement axial [9,129,180]. L'objectif est d'étudier la charge critique de flambement et le comportement post-flambement du panneau.

Dans ce chapitre, la précision des métamodèles est évaluée sur une base de données

de contrôle \mathbf{Q}_c , différente de la base de données d'entraînement \mathbf{Q}_t , au travers de quatre indicateurs, pour chaque sortie α , $\alpha = 1, \dots, M$:

- la valeur de la fonction objectif, avec ou sans terme de lissage, qui transcrit l'efficacité du processus d'optimisation (voir chapitre 3) ;
- le coefficient R_α^2 , qui un indicateur global calculé pour chaque dimension de sortie α par :

$$R_\alpha^2 = 1 - \sum_{i=1}^{n_{\text{TP}}} \frac{(q_{\alpha,i} - h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i}))^2}{(q_{\alpha,i} - \bar{q}_\alpha)^2}, \quad (4.1)$$

avec $q_{\alpha,i}$ les coordonnées des TPs, $h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i})$ les valeurs évaluées par le méta-modèle aux mêmes coordonnées paramétriques $(\zeta_{1,i}, \dots, \zeta_{N,i})$ et \bar{q}_α la valeur moyenne des TPs de la base \mathbf{Q}_c ;

- l'erreur relative¹ moyenne $\bar{\varepsilon}_\alpha$, qui un indicateur global exprimée par :

$$\bar{\varepsilon}_\alpha = \frac{1}{n_{\text{TP}}} \sum_{i=1}^{n_{\text{TP}}} \frac{|q_{\alpha,i} - h_\alpha(\zeta_{1,i}, \dots, \zeta_{N,i})|}{|q_{\alpha,i}|}; \quad (4.2)$$

- la norme \mathcal{L}_1 appliquée à chaque TPs, qui un indicateur local exprimée par :

$$\mathbf{e}_{M_\alpha} = |\mathbf{h}_{\alpha,\text{sep}} - \mathbf{h}_{\alpha,\text{glob}}|. \quad (4.3)$$

4.2 Cas analytiques physiques non-linéaires

4.2.1 Équation de Poisson

Comme illustré dans la Fig. 4.1, les cas d'application traités dans cette première section visent à évaluer la capacité du méta-modèle, basé sur les entités NURBS, à traiter des problèmes non-linéaires. Le premier cas test est faiblement non-linéaire.

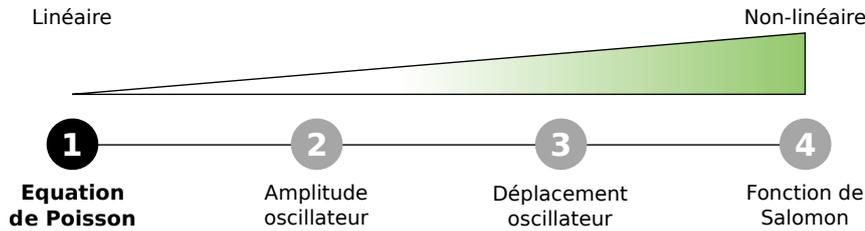


FIGURE 4.1 – Premier cas d'application analytique : équation de Poisson.

4.2.1.1 Description du problème

Le problème de Poisson, tiré de [34], est représentatif des problèmes de propagation des champs de température, i.e., l'équation de la chaleur [95]. Dans sa version la plus

1. La valeur de l'erreur relative moyenne sera systématiquement exprimée en pourcentage pour faciliter sa lecture.

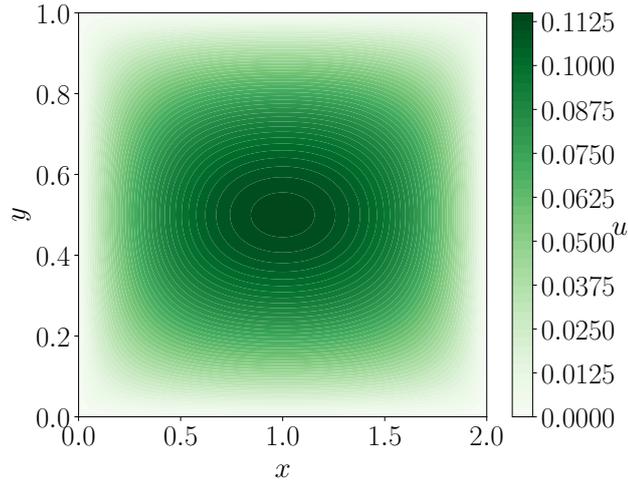


FIGURE 4.2 – Iso-valeurs de $u(x,y)$ représentant la solution analytique du problème de Poisson bidimensionnel exprimée par l’Eq. (4.5).

simple, il est réduit sur un domaine à deux dimensions, avec un terme source constant et unitaire. Initialement, ce problème est employé par [34], pour mettre en évidence les principales caractéristiques de la méthode de réduction de modèle PGD. Plus précisément, ce problème simplifié présente l’efficacité de la méthode PGD à reconstruire une solution inconnue *a priori*. Ici, son comportement faiblement non-linéaire fait un bon premier cas d’application de faible difficulté, pour évaluer l’efficacité de la méthode basée sur les entités NURBS à reproduire un comportement régulier, seulement à partir de points de données, i.e., via l’emploi de la méthode *Centripetal* (voir chapitre 3 et [108]). Les conditions aux limites considérées sont les conditions de Dirichlet. L’équation de Poisson s’exprime alors par :

$$\Delta u(x,y) = f(x,y), \quad (4.4)$$

avec $\Delta u(x,y)$ le Laplacien de la fonction inconnue $u(x,y)$ et $f(x,y) = 1$ le terme source. Les intervalles de variation des deux variables d’entrée sont considérés comme uniformes : $x \in \mathcal{U}[0,2]$ et $y \in \mathcal{U}[0,1]$. La solution exacte de l’Eq. (4.4), tirée de [34], s’exprime par :

$$u_{\text{exact}}(x,y) = \sum_{n,m,\text{odd}} \frac{64}{\pi^4(4n^2 + m^2)} \sin\left(\frac{m\pi x}{2}\right) \sin(n\pi y), \quad (4.5)$$

avec n et m les degrés d’enrichissement de la solution fixés à $n = m = 50$ dans [34]. La Fig. 4.2 illustre cette solution.

4.2.1.2 Caractéristiques et solution du métamodèle

Le problème traité possède $N = 2$ variables d’entrée, (x,y) , et $M = 1$ sortie, u . Pour étudier l’efficacité du métamodèle à reproduire le comportement de l’Eq. 4.5, trois bases

TABLE 4.1 – Caractéristiques des métamodèles employés pour reproduire le comportement de la solution bidimensionnelle de l'équation de Poisson tirée de [34]. Le symbole **M** est employé pour *Métamodèle*.

	Variable	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
M 1			5	4	36	25		
M 2	(x, y)	2	10	9	121	100	$]0, 1[$	$]1, 10[$
M 3			20	19	441	400		
Q_c	(x, y)	(140, 140)			19600			

TABLE 4.2 – Évolution des coefficients f_{obj} , R_α^2 et $\bar{\varepsilon}_\alpha$, en fonction de l'enrichissement de la base \mathbf{Q}_t et de l'étape d'optimisation considérée.

	f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$
M 1	0	0	0	0.9965	0.9965	0.9965	5.3747	5.3747	5.3747
M 2	0	0	0	0.9999	0.9999	0.9999	0.5867	0.5867	0.5867
M 3	0	0	0	1.0000	1.0000	1.0000	0.0984	0.0984	0.0984

de données \mathbf{Q}_t sont considérées, avec trois enrichissement différents renseignés dans le Tab. 4.1. On peut aussi y retrouver les caractéristiques des métamodèles, ainsi que les intervalles de variation des variables d'optimisations β_i et ω_{i_1, \dots, i_N} . Les degrés des polynômes de Bernstein sont fixés à $s_k = 2$ et le rapport τ est défini par $n_k = r_k - 1$ dans la mesure où $r_k + 1 = 5$ dans la plus petite base de données \mathbf{Q}_t (voir conditions sur les dimensions des vecteurs nodaux dans le chapitre 3). En outre, une base \mathbf{Q}_c est employée pour évaluer la justesse des métamodèles avec $r_k = 140$ pour reproduire les conditions de contrôle de précision du métamodèle de [34].

Le calcul des indicateurs de la qualité des métamodèles est effectué pour chaque étape du processus de génération. Les valeurs inférieures à 10^{-16} , sont considérées nulles². Le Tab. 4.2 regroupe ces résultats et la Fig. 4.3 illustre la qualité de l'approximation des différents métamodèles, en traçant les coordonnées des TPs de la base \mathbf{Q}_c en fonction des points évalués par les métamodèles aux mêmes coordonnées paramétriques.

On peut remarquer qu'à partir d'une base \mathbf{Q}_t contenant $n_{\text{TP}} = 441$ TPs, l'erreur relative est inférieure à $\bar{\varepsilon} < 0.1$ % et $R^2 = 1.000$, ce qui traduit une bonne approximation. En considérant seulement $n_{\text{TP}} = 36$, l'erreur relative est de $\bar{\varepsilon} \approx 5$ % et le coefficient de Poisson est de $R^2 = 0.9965$. Ces valeurs restent acceptables et permettent d'utiliser un métamodèle très peu coûteux, constitué de seulement $n_{\text{CP}} = 25$ CPs. Cela correspond également aux informations fournies dans la Fig. 4.3. En outre, les processus d'optimisation relatifs à ξ_2 et ξ_3 sont inefficaces dans cette situation : l'optimisation des CPs est suffisante pour obtenir un métamodèle ayant une très bonne précision.

2. Le zéro numérique de la librairie numpy de Python est de $2.220446049250313 \times 10^{-16}$

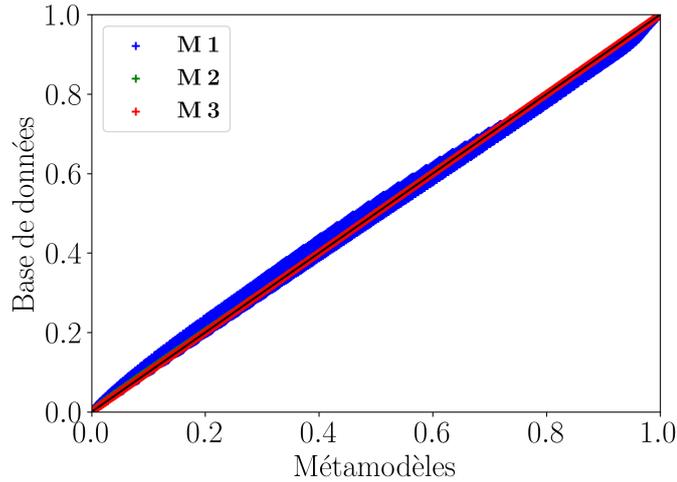


FIGURE 4.3 – Correspondance entre les métamodèles complètement optimisés, respectivement 1, 2 et 3 en bleu, vert et rouge, en fonction des valeurs de la base de données de contrôle.

4.2.2 Système oscillant non-linéaire

Comme illustré dans la Fig. 4.4, le second cas test porte sur l'étude d'un oscillateur. Ce second cas test est faiblement non-linéaire car l'étude se focalise uniquement sur les amplitudes des oscillations.

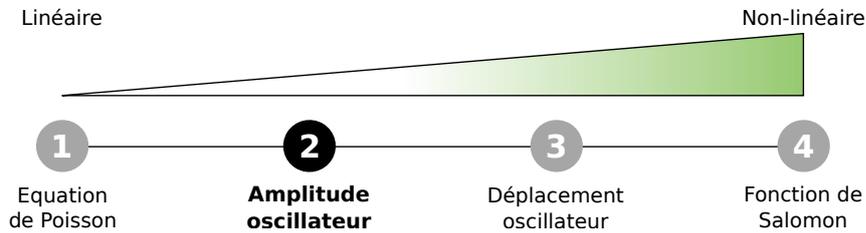


FIGURE 4.4 – Deuxième cas d'application analytique : amplitude maximale d'un système oscillant non-linéaire.

4.2.2.1 Description du problème

Le modèle étudié dans cette section est un oscillateur non-linéaire amorti tiré de [23, 132, 147] et illustré dans la Fig. 4.5. Le but du métamodèle généré dans cette section est d'approximer les amplitudes maximales d'un charriot fixé à un support par deux ressorts et un amortisseur soumis à une impulsion de Dirac. Ses positions extrêmes sont calculées par :

$$x_{\max} = \frac{F_0}{m\omega_0\Lambda}, \quad (4.6)$$

avec m la masse du charriot considérée en glissement sans frottement sur son support, F_0 la valeur de l'impulsion à $t = 0$ s illustrée dans la Fig. 4.5, $\omega_0 = \sqrt{\frac{K}{m}}$ sa pulsation propre, $K = k_1 + k_2$ la raideur totale du système constituée de la somme des raideurs

des deux ressorts et $\Lambda = \frac{c}{2m}$ le coefficient d'amortissement global avec c le coefficient d'amortissement du piston seul. Cette solution analytique permet de construire les bases de données \mathbf{Q}_t et \mathbf{Q}_c .

Le second intérêt de ce problème, outre qu'il soit faiblement non-linéaire, est qu'il possède un nombre de dimensions d'entrées plus élevé que le cas précédent : $N = 5$. La construction des bases de données servant à la génération et au contrôle de la qualité du métamodèle est soumise au fléau des grandes dimensions, i.e., leur taille croît exponentiellement avec leur dimension N (méthode de remplissage de l'espace des possibles en grille uniforme pour un bon recouvrement [72]). La quantité de données à stocker est donc limitée par $r_k < 10$.

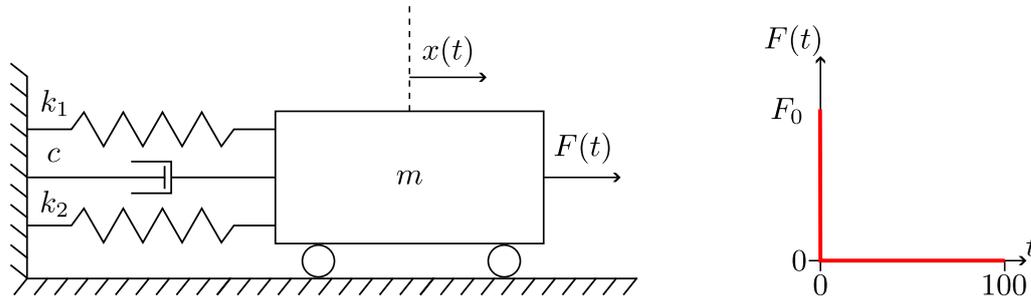


FIGURE 4.5 – A gauche, schéma du dispositif d'oscillation non linéaire inspiré de [23, 132, 147], avec à droite, la représentation du terme source.

4.2.2.2 Caractéristiques et solution du métamodèle

Le métamodèle employé pour approximer cette solution est composé de $N = 5$ dimensions d'entrée, (k_1, k_2, m, c, F_0) , et de $M = 1$ variable de sortie, correspondant à l'amplitude maximale de l'oscillateur, x_{\max} . Les différentes grandeurs caractéristiques du métamodèle sont regroupées dans le Tab. 4.3. Les degrés des polynômes de Bernstein sont pris égaux à $s_k = 2$. La base de données d'entraînement \mathbf{Q}_t est constituée de $r_{\text{TP}} = 32768$ TPs, avec $r_k = 7$ pour garder un nombre de données raisonnable à manipuler. Le nombre de CPs est de $n_{\text{CP}} = 16807$, avec $\tau_k = \frac{2}{3}$. Ainsi, le rapport τ global est de $\tau = \frac{16807}{32768} \approx 0.51$. Cette valeur est comprise dans l'intervalle de variation $\left[\frac{1}{3}, \frac{2}{3}\right]$ définie dans [39], pour un compromis entre précision de résultat, quantité de données à manipuler et minimisation du risque d'*overfitting* (phénomène décrit plus en détail à la section 4.3.2 et au chapitre 3).

Les différentes valeurs des indicateurs pour chaque étape du processus de génération du métamodèle sont regroupées dans le Tab. 4.4. Le processus d'optimisation est efficace et a permis de minimiser la fonction objectif en la faisant passer de $f_{\text{obj}} = 2.04 \times 10^{-6}$ à $f_{\text{obj}} = 1.54 \times 10^{-13}$. Pour un problème à grandes dimensions, avec peu de données d'entraînement, dont le comportement est faiblement non-linéaire, le métamodèle basé sur les entités NURBS est efficace, avec un coefficient final $R_{x_{\max}}^2 = 0.9999$ et une erreur relative moyenne finale $\bar{\varepsilon}_{x_{\max}} = 0.07\%$, les deux quantités étant évaluées sur la base de données de

TABLE 4.3 – Caractéristiques du métamodèle employé pour reproduire l’amplitude du système oscillant non-linéaire.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
k_1	$\mathcal{U}[10, 30]$							
k_2	$\mathcal{U}[0, 20]$							
m	$\mathcal{U}[2, 8]$	2	7	6	32768	16807]0, 1[[1, 10]
c	$\mathcal{U}[1, 4]$							
F_0	$\mathcal{U}[0.5, 1.5]$							
\mathbf{Q}_c		(8, 8, 8, 8, 8)			59049			

 TABLE 4.4 – Évolution des coefficients f_{obj} , R_α^2 et $\bar{\varepsilon}_\alpha$, en fonction de l’étape d’optimisation considérée.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$
1	1.68×10^{-5}	7.55×10^{-8}	0.9999	0.9999	0.9999	0.08	0.07	0.07

contrôle \mathbf{Q}_c . Dans cette situation le terme de lissage n’a pas été employé dans la mesure où le comportement du métamodèle était très bon avec le paramétrage choisi.

4.2.3 Système oscillant non-linéaire : évolution temporelle

Comme illustré dans la Fig. 4.6, ce troisième cas test est fortement non-linéaire car il vise à reproduire la réponse de l’oscillateur dans l’intervalle temporel considéré.

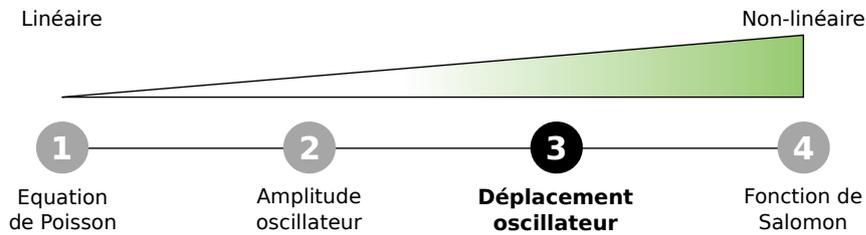


FIGURE 4.6 – Premier cas d’application analytique : position au cours du temps d’un système oscillant non-linéaire.

4.2.3.1 Description du problème

Le problème traité dans cette section est le même que celui traité à la section précédente mais en considérant la réponse dynamique de l’oscillateur et non seulement son amplitude maximale. L’intérêt de ce cas d’application réside dans son comportement non-linéaire au cours du temps. Cela implique de considérer un grand nombre de données dans la direction d’entrée correspondante. Néanmoins, ce problème possède, comme le précédent, $N = 5$ dimensions d’entrée (la masse est fixée à $m = 1$ kg et est remplacée par la variable temporelle t). La problématique est alors la suivante : le métamodèle peut-il reproduire les non-linéarités temporelles avec un faible nombre de données ?

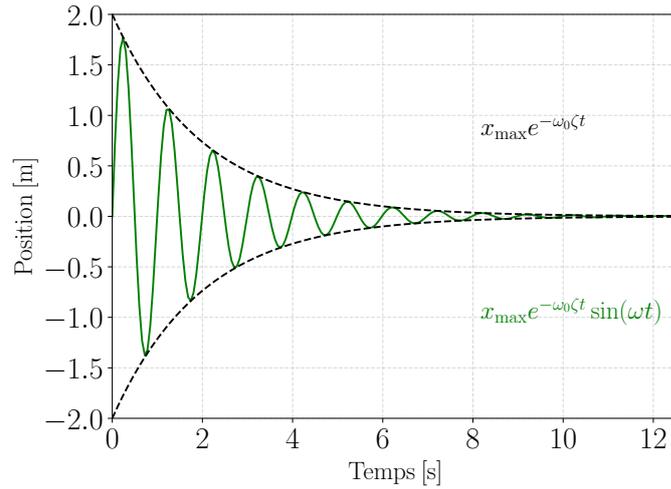


FIGURE 4.7 – Évolution au cours du temps de la position de la masse pour le jeu de paramètres suivant : $m = 1$ kg, $k_1 = 25$ N.m⁻¹, $k_2 = 15$ N.m⁻¹, $c = 1$ N.m⁻¹.s⁻¹ et $F_0 = 1$ N. En vert, la position de la masse et en noir son enveloppe d’amortissement.

Le comportement dynamique de ce dispositif est déterminé par la résolution d’une équation différentielle du second ordre à coefficients constants, donnée par :

$$\ddot{x}(t) + 2\Lambda\dot{x}(t) + \omega_0^2 x(t) = F(t). \quad (4.7)$$

Le problème est considéré pseudo-périodique ($\Lambda^2 - \omega_0^2 < 0$) soumis à une impulsion de Dirac F_0 , sans déphasage initial, $\phi = 0$. La solution de l’Eq. (4.7) est alors exprimée par :

$$x(t) = x_{\max} e^{-\omega_0 \zeta t} \sin(\omega t), \quad (4.8)$$

avec $w = \omega_0 \sqrt{1 - \zeta^2}$ la pulsion de l’oscillateur. Les bases de données \mathbf{Q}_t et \mathbf{Q}_c sont construites à partir de cette équation analytique. La Fig. 4.7 représente, en vert, la position de la masse au cours du temps pour le jeu de paramètres suivant : $m = 1$ kg, $k_1 = 25$ N.m⁻¹, $k_2 = 15$ N.m⁻¹, $c = 1$ N.m⁻¹.s⁻¹ et $F_0 = 1$ N, et en noir l’enveloppe de l’amortissement.

4.2.3.2 Caractéristiques et solution du métamodèle

Les différentes grandeurs caractéristiques du métamodèle sont données dans le Tab. 4.5. Le degré des polynômes de Bernstein sont pris égaux à $s_k = 3$ compte tenu du comportement non-linéaire du dispositif, illustré par la Fig. 4.7. Le nombre de CPs est déterminé via le rapport τ en fonction de $n_k = \left(\frac{2}{3}\right)^{\frac{1}{5}} r_k$. Ainsi, le rapport τ est de $\tau = \frac{n_{\text{CP}}}{r_{\text{TP}}} \approx 0.53$. Quant à la base de données de contrôle, elle est constituée de $n_{\text{TP}} = 126976$ TPs.

L’évaluation de l’efficacité des optimisations est faite via la fonction objectif f_{obj} . La justesse du métamodèle est déterminée par les indicateurs R_α^2 et $\bar{\varepsilon}_\alpha$, les deux quantités étant évaluées sur la base de contrôle \mathbf{Q}_c . Le Tab. 4.6 regroupe ces résultats et la Fig. 4.8

TABLE 4.5 – Caractéristique du métamodèle pour reproduire le déplacement du système oscillant non-linéaire.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
k_1	$\mathcal{U}[10, 30]$							
k_2	$\mathcal{U}[0, 20]$							
c	$\mathcal{U}[1, 4]$	3	7	6	126976	67228]0, 1[[1, 10]
F_0	$\mathcal{U}[0.5, 1.5]$							
t	$\mathcal{U}[0, 4\pi]$		30	27				
\mathbf{Q}_c		(8, 8, 8, 8, 31)			209952			

 TABLE 4.6 – Évolution des coefficients f_{obj} , R_α^2 et $\bar{\varepsilon}_\alpha$, en fonction de l'étape d'optimisation considérée.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$
1	2.47×10^{-2}	2.37×10^{-2}	0.9890	0.9894	0.9894	3.94	3.66	3.62

illustre la qualité de l'approximation des différents métamodèles, en traçant les TPs de la base de données de contrôle en fonction des points évalués par le métamodèle aux mêmes coordonnées paramétriques.

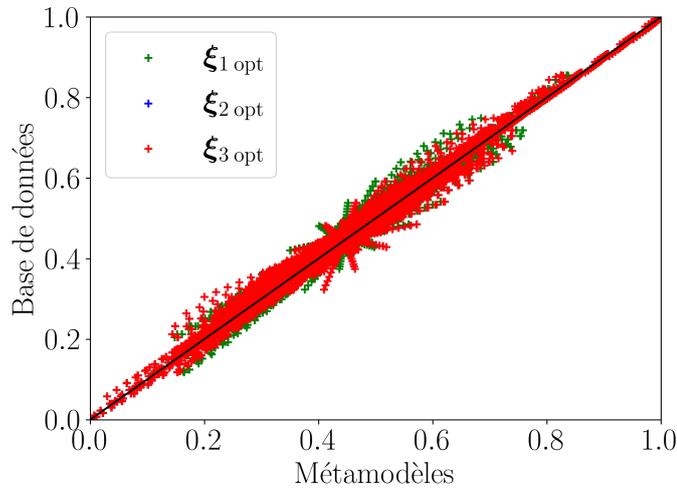


FIGURE 4.8 – Graphique traçant les coordonnées des TPs de la base de données de contrôle en fonction des résultats fournis par le second métamodèle du système oscillant non-linéaire. Le comportement du métamodèle au cours des trois étapes d'optimisation est représenté par les croix bleues, vertes et rouges respectivement.

Suite à l'analyse de ces résultats, on peut déduire que le métamodèle est capable de reproduire l'évolution temporelle du système en capturant les non-linéarités avec une bonne précision. Plus précisément, dès la première étape du processus d'optimisation, les différents indicateurs ont des valeurs acceptables. Cependant, la valeur de la fonction de coût reste encore élevée et une amélioration est encore possible en optimisant la valeur des

composants des vecteurs nodaux. En effet, à la fin de la seconde étape du processus d'optimisation la fonction objectif passe de 1 à 2.47×10^{-2} ce qui représente une réduction d'un facteur 100. Comme cela expliqué au chapitre 3, le rôle principal de cette deuxième phase d'optimisation est de positionner correctement les composantes non-triviales des vecteurs nodaux afin de capturer correctement les non-linéarités. La dernière étape d'optimisation portant sur les poids, est quant à elle peu efficace : ses indicateurs de précision sont très proches de ceux issus de la seconde étape d'optimisation.

4.2.4 Propagation d'une onde à la surface d'un fluide

Comme illustré dans la Fig. 4.9, ce quatrième et dernier cas test analytique est très fortement non-linéaire, avec une haute fréquence de vibration (impact à la surface d'un fluide).

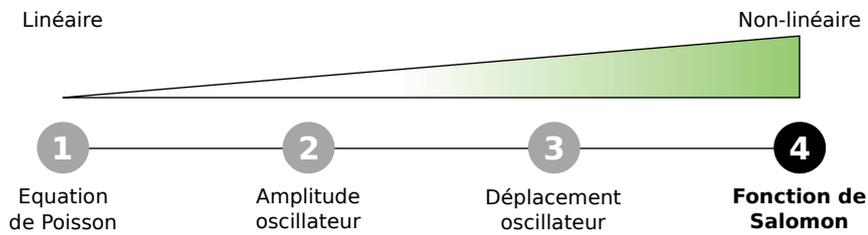


FIGURE 4.9 – Quatrième cas d'application analytique : la fonction de Salomon.

4.2.4.1 Description du problème

Le problème étudié dans cette section correspond à la fonction de Salomon [63, 150] qui traduit le mouvement de la surface d'un fluide soumis à une impulsion de Dirac en son centre. Il s'agit de la solution analytique de la propagation d'une onde acoustique de forte magnitude à la surface d'un fluide, illustrée par la Fig. 4.10. Initialement employée dans le domaine de l'optimisation [150] pour sa particularité à posséder un grand nombre de minimum locaux en plus d'un minimum global, elle peut être étudiée dans le domaine de la métamodélisation pour son aspect fortement non-linéaire [63]. C'est cette dernière particularité qui rend intéressant ce cas d'application, dans la mesure où ce comportement non-linéaire est très important, pouvant être considéré comme du bruit à minimiser dans d'autres situations.

Cette fonction est étudiée dans la chapitre 3 pour illustrer le rôle de chaque optimisation du processus de génération du métamodèle. Les différents indicateurs R^2 , $\bar{\varepsilon}$ et E_M sont alors calculés entre le métamodèle et la base de données d'entraînement \mathbf{Q}_t . Dans cette nouvelle étude, le but est de reproduire le comportement de la fonction sur des points différents de sa base d'entraînement. Pour rappel, la fonction de Salomon s'exprime par :

$$u(x_1, x_2) = 1 - \cos\left(2\pi\sqrt{x_1^2 + x_2^2}\right) + 0.1\sqrt{x_1^2 + x_2^2}, \quad (4.9)$$

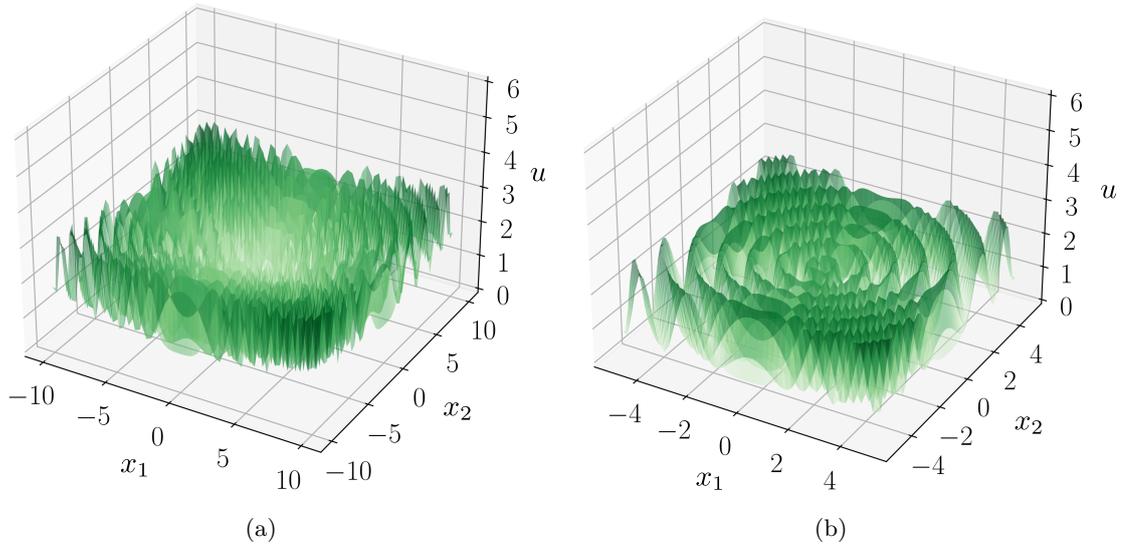


FIGURE 4.10 – Représentation de la propagation de l’onde de choc à la surface d’un fluide, initiée par une impulsion de Dirac. Les intervalles de variations des deux variables d’entrée sont définis sur $\mathcal{U}[-10, 10]$, puis réduits sur $\mathcal{U}[-5, 5]$ pour une meilleure visualisation.

avec $(x_1, x_2) \in \mathcal{U}[-10, 10]$ les coordonnées de l’espace des entrées. La Fig. 4.10 illustre cette équation avec un diagramme évalué sur le domaine $(x_1, x_2) \in \mathcal{U}[-10, 10]$ et un autre sur le domaine $(x_1, x_2) \in \mathcal{U}[-5, 5]$ pour une meilleure visualisation.

4.2.4.2 Caractéristiques et solution du métamodèle

Le métamodèle employé pour approximer le comportement du problème défini par l’Eq. (4.9) est composé de $N = 2$ dimensions d’entrée et de $M = 1$ dimension de sortie. Le phénomène d’*overfitting* apparaît quand le nombre de CPs est proche du nombre de TPs. Pour le limiter, il est nécessaire d’employer le terme de lissage (voir chapitre 3). Cependant, ce terme va avoir tendance à lisser les non-linéarités qui, pour ce modèle, doivent toutes être prises en compte : il ne sera pas pris en compte dans la suite. Les degrés des polynômes de Bernstein sont alors fixés à $s_k = 2$ et non à $s_k = 3$ pour limiter ce phénomène. En outre, plusieurs métamodèles sont appliqués sur ce problème pour proposer de déterminer la meilleure configuration : vaut-il mieux augmenter le nombre de TPs ou le rapport τ en présence de fortes non-linéarités ? Le Tab. 4.7 rassemble les valeurs des variables globales des métamodèles étudiés : trois tailles de base \mathbf{Q}_t et trois rapports τ sont considérées.

Le Tab. 4.8 regroupe les valeurs des différents indicateurs définis plus haut et les Figs. 4.11 et 4.12 illustrent la qualité de l’approximation, en traçant les coordonnées des TPs de la base de données de contrôle en fonction des points évalués par les métamodèles optimisés, aux mêmes coordonnées paramétriques, en considérant respectivement $\tau = \frac{2}{3}$ pour les trois bases d’entraînement et $r_k + 1 = 64$ pour les trois valeurs de τ étudiées.

TABLE 4.7 – Caractéristiques des métamodèles employés pour reproduire le comportement de l’onde de choc à la surface d’un fluide [150].

	Variable	s_k	r_k	n_k	τ_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
M 1	(x_1, x_2)	2	53	43	$(\frac{2}{3})^{1/2}$	2916	1936]0, 1[[1, 10]
M 2				47	$(\frac{4}{5})^{1/2}$		2304		
M 3				52	$(\frac{52}{53})^{1/2}$		2809		
M 4	(x_1, x_2)	2	63	51	$(\frac{2}{3})^{1/2}$	4096	2704]0, 1[[1, 10]
M 5				56	$(\frac{4}{5})^{1/2}$		3249		
M 6				59	$(\frac{62}{63})^{1/2}$		3969		
M 7	(x_1, x_2)	2	73	59	$(\frac{2}{3})^{1/2}$	5476	3600]0, 1[[1, 10]
M 8				65	$(\frac{4}{5})^{1/2}$		4356		
M 9				72	$(\frac{72}{73})^{1/2}$		5484		
Q_c	(x_1, x_2)	(140, 140)				19881			

TABLE 4.8 – Évolution des coefficients f_{obj} , R_α^2 et $\bar{\varepsilon}_\alpha$, en fonction de l’enrichissement de la base de données d’entraînement et de l’étape d’optimisation considérée.

	f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$
M 1	1	0.65	0.42	0.8395	0.8503	0.8624	18.27	17.54	17.18
M 2	1	0.20	0.06	0.9072	0.8957	0.8957	13.77	16.40	16.52
M 3	1	0.22	0.13	0.9466	0.9466	0.9466	10.94	10.93	10.93
M 4	1	0.36	0.17	0.9584	0.9571	0.9585	9.29	9.62	9.31
M 5	1	0.03	0.003	0.9715	0.9632	0.9629	7.62	8.44	8.49
M 6	0	0	0	0.9852	0.9852	0.9852	5.76	5.76	5.76
M 7	1	0.11	0.03	0.9904	0.9860	0.9856	4.57	5.40	5.45
M 8	1	0.01	3.9×10^{-4}	0.9933	0.9927	0.9920	3.61	3.83	3.84
M 9	0	0	0	0.9966	0.9966	0.9966	2.65	2.65	2.65

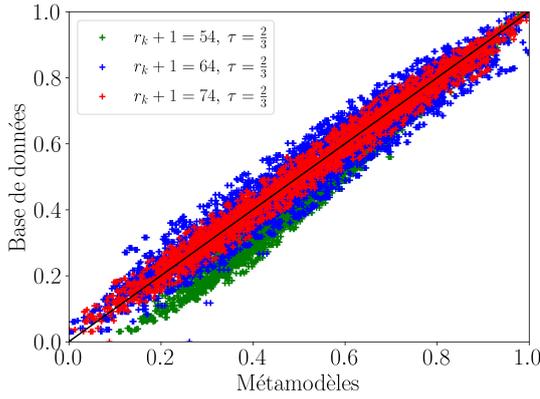


FIGURE 4.11 – Coordonnées des *Target Points* de la base de données de contrôle en fonction des résultats calculés par les différents métamodèles pour trois enrichissements différents avec $\tau = \frac{2}{3}$.

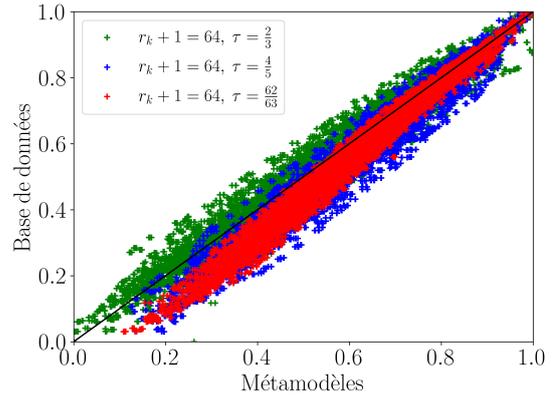


FIGURE 4.12 – Coordonnées des *Target Points* de la base de données de contrôle en fonction des résultats calculés par les différents métamodèles pour trois valeurs de τ différents pour $r_k + 1 = 64$.

Tout comme la situation illustrée au chapitre 3, la valeur de la fonction objectif diminue, ou reste constante quand sa valeur est très basse dès la première étape d'optimisation, i.e., $f_0 < 10^{-4}$. Cela met en évidence l'efficacité du processus d'optimisation à rapprocher le métamodèle des données de la base \mathbf{Q}_t . Par opposition à la situation du chapitre 3, les valeurs des indicateurs R_α^2 et $\bar{\varepsilon}_\alpha$ par rapport à une base \mathbf{Q}_c n'ont pas un comportement trivial. Pour le métamodèle **M 1**, la valeur de R_α^2 augmente et celle de $\bar{\varepsilon}_\alpha$ diminue, contrairement au métamodèle **M 5**. Ce comportement met en évidence une limitation de cette méthode : si les non-linéarités du modèle étudié ne sont pas prises en compte dans la base d'entraînement, le métamodèle ne peut pas les reproduire correctement. En outre, si ces non-linéarités sont trop fortes et en trop grand nombre, le processus d'optimisation va permettre un bon *fitting* de la base \mathbf{Q}_t au détriment du *fitting* de la base \mathbf{Q}_c . Pour avoir une bonne approximation, la solution est alors d'augmenter le nombre de TPs de \mathbf{Q}_t : une réflexion est alors nécessaire sur la meilleure façon d'employer le métamodèle sur un cas aussi non-linéaire, où le métamodèle pourrait être mis à défaut, tout comme toute autre solution d'approximation. Néanmoins, les Figs. 4.11 et 4.12 montrent que les différents métamodèles, après optimisation, reproduisent l'allure générale de la fonction, même dans les situations critiques ce qui illustre la performance d'une métamodélisation basée sur une hypersurface NURBS.

4.3 Problèmes mécaniques nécessitant une résolution numérique

4.3.1 Plaque pleine

Comme illustré dans la Fig. 4.13, les cas d'application traités dans cette première section visent à évaluer la capacité du métamodèle basé sur les hypersurfaces NURBS à traiter non seulement des non-linéarités, mais aussi à s'adapter à des complexités liées à des singularités, à un faible nombre de données d'entraînement ou encore au besoin d'étudier un champ. Ce premier cas test est faiblement non-linéaire, mais met en œuvre une étude de champ de déplacements.

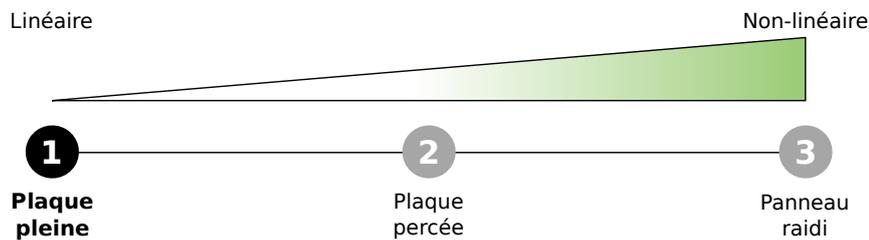


FIGURE 4.13 – Premier cas d'application analytique : plaque pleine.

4.3.1.1 Description du problème

Ce cas test correspond à l'étude du champ de déplacement en grande déformation d'une plaque mince, pleine encastree en flexion. Le problème est schématisé dans la Fig. 4.14a. La force de flexion, constante, est non-linéaire dans la mesure où sa norme suit la déformation de la plaque. Le matériau constitutif de la plaque est bi-linéaire élastique parfaitement plastique. Ses propriétés sont tirées de [7] et listées dans le Tab. 4.9. La modélisation est réalisée par une analyse non-linéaire via le logiciel ANSYS/APDL en considérant un maillage de 1250 éléments SHELL181 avec quatre nœuds et six degrés de liberté par nœud. Le maillage et sa structure, sont représentés par la Fig. 4.14b. Ce cas d'application est intéressant bien qu'il soit faiblement non-linéaire, dans la mesure où il requiert une approximation d'un champ de déplacements.

Les bases de données d'entraînement \mathbf{Q}_t et de contrôle \mathbf{Q}_c sont générées via l'analyse EF de ce modèle en considérant $N = 4$ variables d'entrée et $M = 1$ sortie. Cette dernière correspond au déplacement de la plaque selon l'axe z . Quant aux variables d'entrée, elles comprennent la force de flexion F , l'épaisseur de la plaque e_{plq} et les coordonnées (x, y) auxquelles sont évalués les déplacements u_z , représentées par la grille de la Fig. 4.14a.

4.3.1.2 Caractéristiques et solution du métamodèle

Le métamodèle employé pour approximer le champ de déplacements de cette plaque en flexion est constitué de $N = 4$ variables d'entrée, $(F, e_{\text{plq}}, x, y)$, et de $M = 1$ sortie, u_z .

TABLE 4.9 – Caractéristiques du matériau employé dans l’analyse EF des plaques en flexion tirées de [7].

Grandeur	Valeur
Module d’Young [Pa]	200×10^9
Coefficient de Poisson [-]	0.3
Contrainte seuil [Pa]	200×10^6

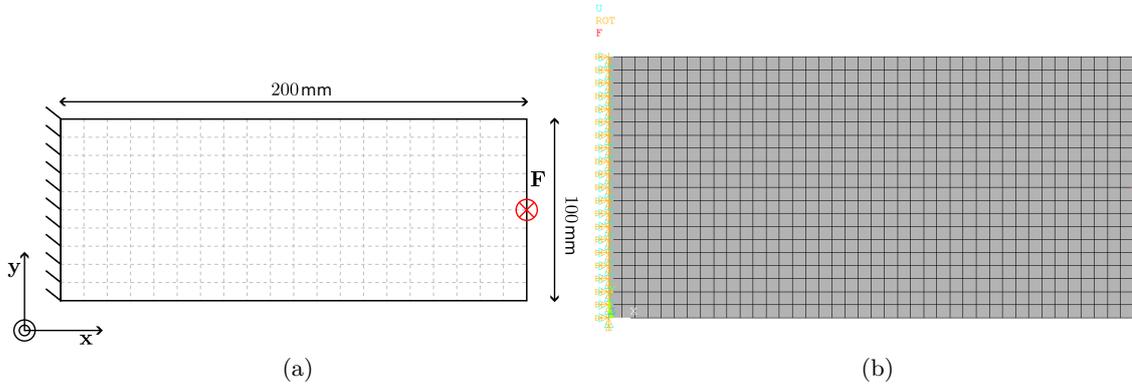


FIGURE 4.14 – Schéma de la plaque pleine soumise à une force de flexion en (a) et son maillage sous ANSYS/APDL en (b).

Les caractéristiques de ce métamodèle sont renseignées dans le Tab. 4.10 tout comme les intervalles de variations des variables d’entrée et des variables d’optimisation, i.e., les coefficients β_i et les poids ω_{i_1, \dots, i_N} . Le degré des polynômes de Bernstein est fixé à $s_k = 2$ et le rapport entre les TPs et les CPs est de $\tau = \frac{2}{3}$. Les grandeurs n_k et r_k pour les bases de données d’entraînement et de contrôle sont également reportées dans le Tab. 4.10.

L’efficacité des différentes étapes d’optimisations est mesurée via l’évolution de la fonction objectif f_{obj} . Quant à la précision du métamodèle, elle est évaluée par l’erreur relative $\bar{\varepsilon}_\alpha$ et le coefficient R_α^2 , mesurés sur la base de contrôle \mathbf{Q}_c . Ces résultats sont renseignés dans le Tab. 4.10.

Dès la première étape du processus d’optimisation, on peut remarquer que les valeurs de la fonction objectif et des indicateurs R^2 et $\bar{\varepsilon}$ sont largement acceptables, avec $R^2 = 0.9999$ et $\bar{\varepsilon} = 0.88\%$. Ces résultats mettent en avant l’efficacité du métamodèle à reproduire un champ de déplacement faiblement non-linéaire avec des bases de données réduites pour

TABLE 4.10 – Caractéristiques du métamodèle pour reproduire le comportement de la plaque pleine en flexion.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
x	$\mathcal{U}[0, 200]$		20	18				
y	$\mathcal{U}[0, 100]$		10	9				
F	$\mathcal{U}[150, 300]$	2	5	4	8316	5000	$]0, 1[$	$[1, 10]$
e_{plq}	$\mathcal{U}[1, 10]$		5	4				
\mathbf{Q}_c			(19, 9, 13, 13)		39200			

son entraînement et son contrôle en précision. En outre, le gain en temps de simulation est important : environ 120 s sont nécessaires pour mener à bien une analyse EF de ce problème contre moins de 1 s par le métamodèle. Il convient de noter que la construction de la base de données d'entraînement employée dans ce cas d'application requiert environ 4440 s pour être générée.

TABLE 4.11 – Évolution des coefficients f_{obj} , R_1^2 et $\bar{\varepsilon}_1$, en fonction de l'enrichissement de la base de données d'entraînement et de l'étape d'optimisation considérée.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$
1	0.31	0.30	0.9999	0.9999	0.9999	0.88	0.83	0.83

4.3.2 Plaque percée

Comme illustré dans la Fig. 4.15, le second cas test est non-linéaire, requiert une étude de champ et possède une singularité importante dans sa géométrie.

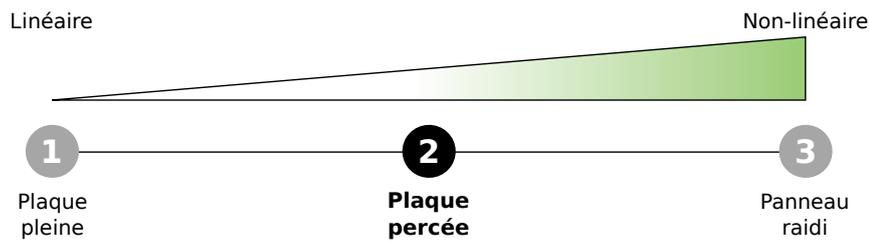


FIGURE 4.15 – Second cas d'application analytique : plaque percée.

4.3.2.1 Description du problème

Ce cas test vise à approximer le champ de déplacement en grande déformation d'une plaque mince percée, en flexion. Ce problème est schématisé dans la Fig. 4.16a. La force de flexion est constante mais change en direction dans la mesure où elle reste normale à la plaque tout au long de son déplacement. Le matériau constitutif de la plaque est le même que pour le cas précédent (voir Tab. 4.9). Le modèle aux EF consiste en un maillage de 5650 éléments SHELL181, avec quatre nœuds et six de degrés de liberté par nœud. Le maillage utilisé est illustré dans la Fig. 4.16b et a fait l'objet d'une étude de convergence non reportée ici par soucis de concision. Ce cas d'application est intéressant, dans la mesure où il requiert une approximation d'un champ de déplacements avec une singularité géométrique.

Les NURBS étant des entités géométriques continues par définition, le métamodèle construit à partir de ces entités n'est pas particulièrement adapté à ce type de problème. Néanmoins des solutions existent :

- considérer un déplacement nul dans la zone de singularité. Ainsi, l’hypersurface continue pourra approximer le comportement de l’ensemble. Cette piste a commencé à être explorée par [11] ;
- diviser le domaine discontinu en plusieurs domaines continus puis les raccorder selon des conditions de continuité sur les dérivées des hypersurfaces [18] ;
- opérer une transformation de l’espace pour le rendre continu. Dans cette situation cela revient à « dérouler » la plaque autour du trou. Cette méthode a été explorée par [3].

Dans le cadre de ce travail, nous avons exploité la première solution. Les bases de données sont construites en évaluant les déplacements de la plaque à des coordonnées différentes de celles des nœuds du maillage, représentées par la grille de la Fig. 4.16a.

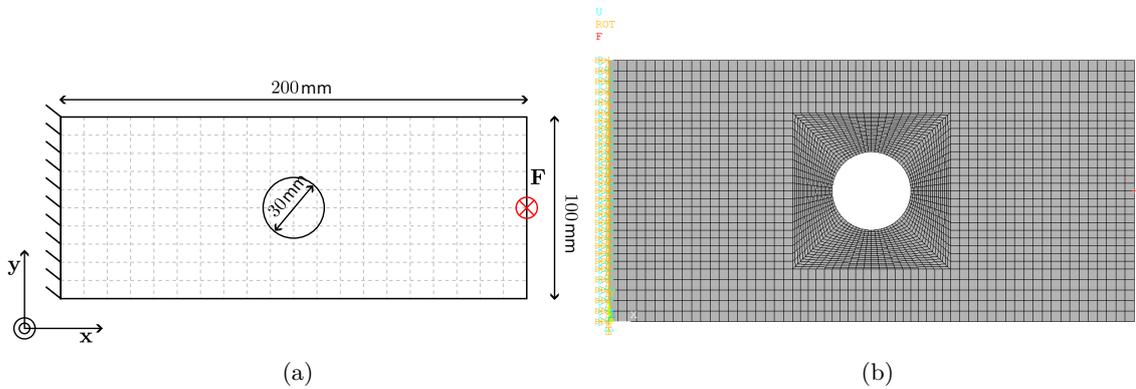


FIGURE 4.16 – Schéma de la plaque percée soumise à une force de flexion en (a) et son maillage sous ANSYS/APDL en (b).

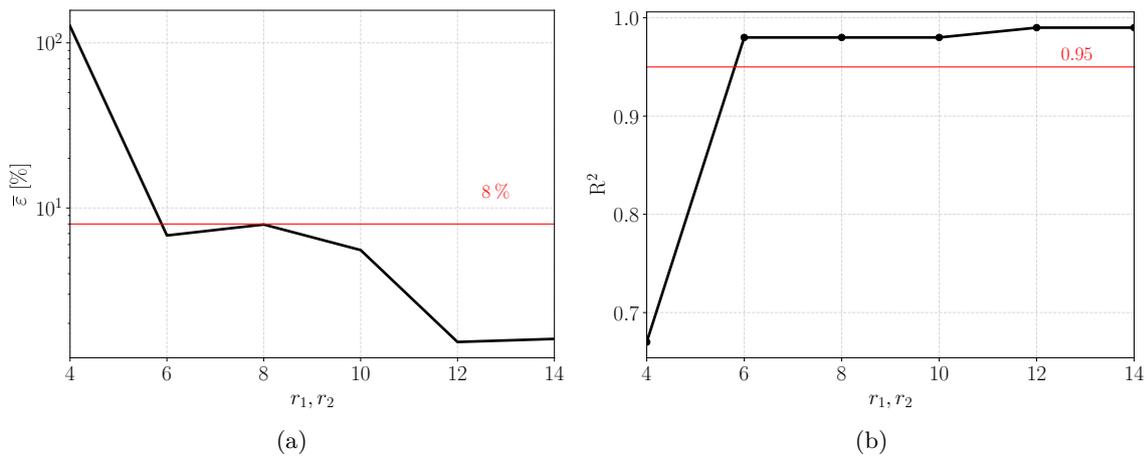
4.3.2.2 Caractéristiques et solution du métamodèle

Le métamodèle employé pour approximer le champ de déplacements en grande déformation de cette plaque est constitué de $N = 4$ variables d’entrée, $(F, e_{\text{plq}}, x, y)$ et $M = 1$ sortie, u_z . Les caractéristiques de ce métamodèle sont renseignées dans le Tab. 4.13, tout comme les intervalles de variation des variables d’entrée et des variables d’optimisation, i.e., β_i et ω_{i_1, \dots, i_N} . Les degrés des polynômes de Bernstein sont fixés à $s_k = 2$. La singularité de ce problème s’apparente à la singularité de l’exemple SIMO du créneau du chapitre 3 mais en plus grande dimension. Ainsi, le rapport $\tau_k = \left(\frac{2}{3}\right)^{1/4}$ pour $k = 1, 2$, F et e_{plq} respectivement, et $r_k = n_k - 1$ pour $k = 4, 5$, pour x et y respectivement. Ce choix est fait pour une bonne approximation, en plus de considérer le terme de lissage pour éviter tout problème d’*overfitting*. En guise d’illustration de ce phénomène, le cas d’application de la plaque percée sans prise en compte de ce terme est proposé en annexe D.1.

Avant d’entamer la procédure de génération du métamodèle, une étude de convergence a été faite sur la quantité de données à prendre en compte dans la base \mathbf{Q}_t . Les diffé-

TABLE 4.12 – Caractéristiques des bases de données d’entraînement et de contrôle utilisées pour l’étude de convergence.

	r_1	r_2	r_3	r_4	n_{TP}
$\mathbf{Q}_{t,1}$	4	4			10150
$\mathbf{Q}_{t,2}$	6	6			19894
$\mathbf{Q}_{t,3}$	8	8	28	13	32886
$\mathbf{Q}_{t,4}$	10	10			49126
$\mathbf{Q}_{t,5}$	12	12			68614
$\mathbf{Q}_{t,6}$	14	14			91350
\mathbf{Q}_c	4	4	31	16	10150

FIGURE 4.17 – Étude de convergence sur l’enrichissement de la base d’entraînement en considérant $\bar{\varepsilon}_\alpha$ en (a) et R_α^2 en (b).

rentes valeurs r_k liées aux dimensions des bases sont renseignées dans le Tab. 4.12. Plus précisément, les coordonnées cartésiennes auxquelles sont évalués les déplacements u_z sont constantes tout au long de cette étude. Seules r_1 et r_2 sont enrichies.

Les études de convergence (Fig. 4.17), montrent qu’il est nécessaire de considérer au moins 68614 (**DB 5**) TPs pour atteindre la convergence en termes de précision. En conséquence, les valeurs des variables de conception après enrichissement sont renseignées dans le Tab. 4.13. L’efficacité de l’optimisation basée sur le gradient est mesurée par l’évolution de la fonction objectif f_{obj} , tandis que la précision du métamodèle est évaluée par l’erreur

TABLE 4.13 – Caractéristiques du métamodèle appliqué sur le problème de la plaque percée.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
x	$\mathcal{U}[0, 200]$		28	27				
y	$\mathcal{U}[0, 100]$		13	12				
F	$\mathcal{U}[150, 300]$	2	12	8	68614	29484	$]0, 1[$	$[1, 10]$
e_{plq}	$\mathcal{U}[4, 14]$		12	8				
\mathbf{Q}_c			(31, 16, 6, 6)		26656			

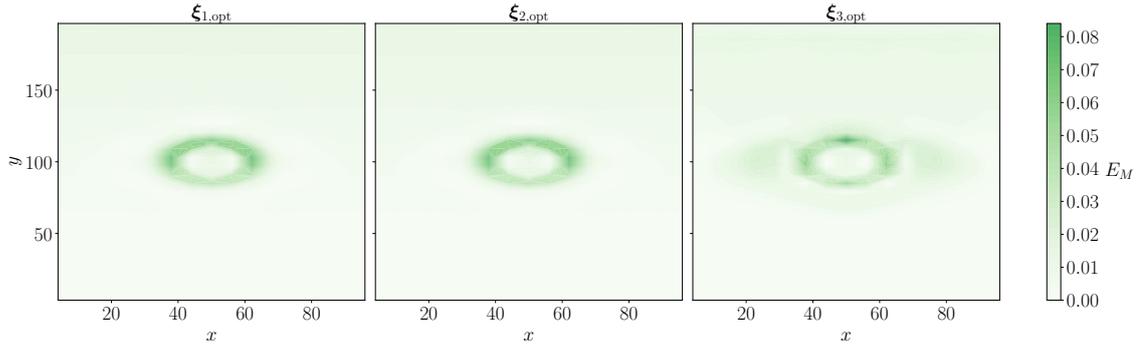


FIGURE 4.18 – Iso-valeurs des écarts e_M entre la base de données de contrôle et le méta-modèle considéré aux différentes étapes de sa génération. $F = 173.7$ N et $e_{\text{plq}} = 5.57$ mm.

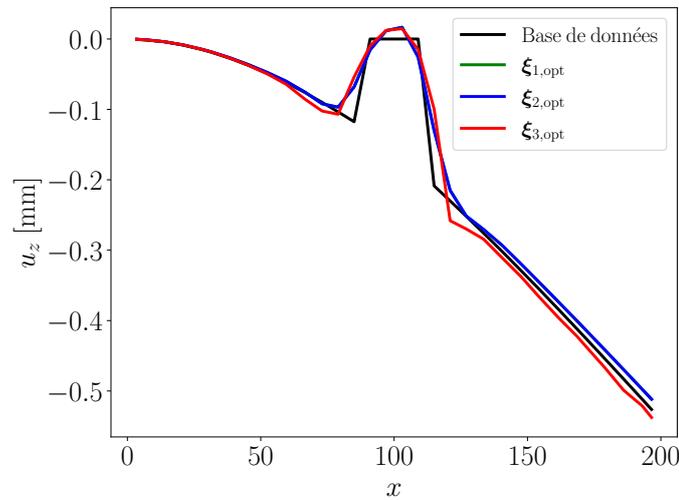


FIGURE 4.19 – Champ de déplacement u_z évalué pour $F = 173.7$ N, $e_{\text{plq}} = 5.57$ mm et $y = 50$ mm en fonction de l'étape de génération du métamodèle.

relative $\bar{\varepsilon}_\alpha$ et le coefficient R_α^2 , sur la base de données de contrôle \mathbf{Q}_c illustrée dans le Tab. 4.13. Ces résultats sont donnés dans le Tab. 4.14.

TABLE 4.14 – Évolution des coefficients f_{obj} , R_1^2 et $\bar{\varepsilon}_1$, en fonction de l'enrichissement de la base de données d'entraînement et de l'étape d'optimisation considérée.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$
1	0.02	0.007	0.9956	0.9956	0.9975	7.3	7.3	6.7

Pour mieux visualiser l'efficacité de la stratégie de métamodélisation développée dans ce travail, les courbes de niveau associées à e_M sont tracées dans la Fig. 4.18. En outre, le champ de déplacement u_z en fonction de la coordonnée x et pour des valeurs données de force F , épaisseur e_{plq} et coordonnée y pour les trois étapes d'optimisation est illustrée dans la Fig. 4.19. De la même manière que pour le cas du créneau du chapitre 3, on observe

une augmentation légère de l'erreur relative et une diminution du coefficient R_α^2 au cours des différentes étapes du processus de génération du métamodèle. Quant à la valeur de la fonction de coût, elle décroît. Cela traduit une bonne optimisation tout en limitant les problèmes d'*overfitting*.

4.3.3 Panneau raidi en flambement

Comme illustré dans la Fig. 4.20, le modèle initial aux EF est coûteux à mettre en place et à coupler avec d'autres modélisations dans le contexte de la modélisation d'un système complet.

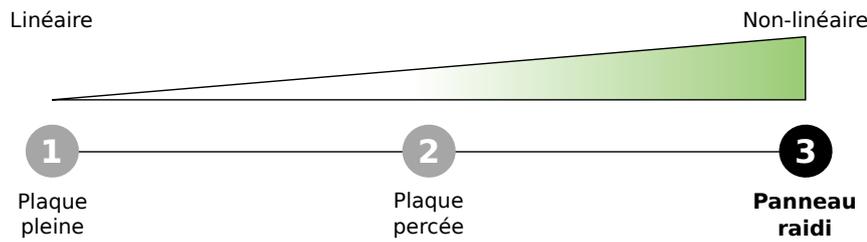


FIGURE 4.20 – Troisième cas d'application analytique : panneau raidi.

4.3.3.1 Description du problème

Dans l'aéronautique, les structures en jeu sont souvent soumises à des contraintes importantes. Néanmoins, pour limiter tant les coûts de production que d'utilisation, les matériaux constitutifs doivent être les plus légers possible. Il est donc important de trouver des compromis entre tenues mécaniques et masses. Un des phénomènes les plus redouté est le flambement : tendance d'un matériau à fléchir et à se déformer perpendiculairement à la force appliquée. Cette déformation est difficilement prévisible. L'objectif de cette section est d'appliquer la stratégie de métamodélisation développée dans ce travail sur un motif répété de panneau raidi soumis à une charge axiale.

La géométrie du modèle d'étude ainsi que les paramètres associés sont tirés de [129] et illustrés dans la Fig. 4.21. Ce panneau raidi est constitué de deux parties : le panneau et le raidisseur. Le panneau, de longueur a et de largeur $b = 2(a_1 + a_2 + a_3)$, est constitué de n_S plis pré-imprégnés de carbone-époxy dont les propriétés sont données par [129]. Le raidisseur est constitué de n_B plis et sa géométrie est définie par les paramètres a_2 , a_3 et h de la Fig. 4.21. Pour concevoir ce panneau les relations suivantes sont introduites :

$$c_1 = 2\frac{a_2}{a}, c_2 = 2\frac{a_3}{a_2}, c_3 = \frac{h}{a_2}. \quad (4.10)$$

Pour éviter de générer des géométries incohérentes, deux inégalités sont introduites comme suit :

$$a_1 = \frac{a}{2} - a_2 - a_3, \theta = \text{atan}\left(\frac{h}{a_3 - \frac{a_2}{2}}\right), \text{ avec } a_1 > 0, a_3 \geq \frac{a_2}{2}. \quad (4.11)$$

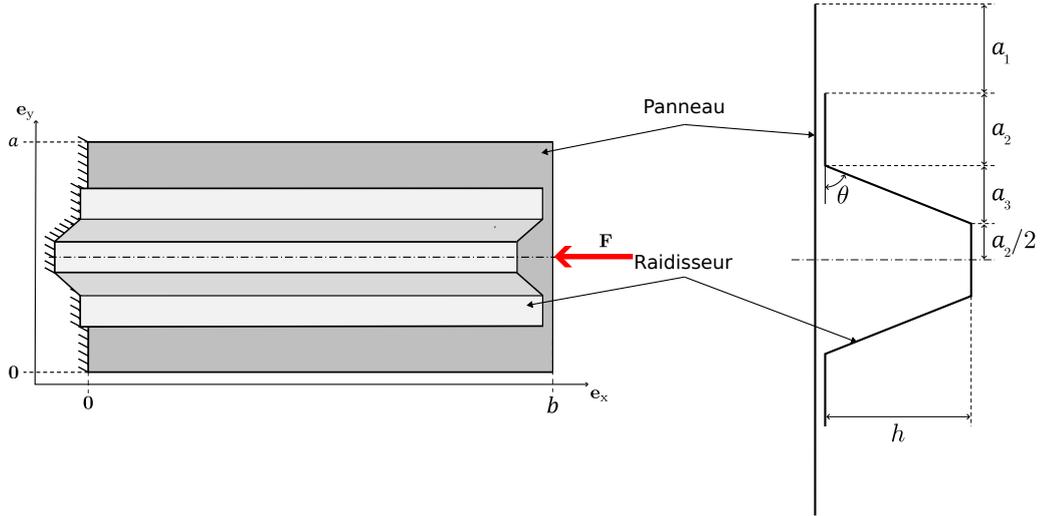


FIGURE 4.21 – Schéma du motif élémentaire formant le panneau raidi. Application de la force axiale en rouge.

Outre ses paramètres géométriques, cette structure est également décrite par ses paramètres matériaux, définis par le formalisme polaire, qui est une approche efficace pour définir la réponse mécanique des stratifiés composites [93, 124]. Dans cette situation, seuls trois paramètres polaires sont nécessaires pour décrire le comportement élastique macroscopique d'un stratifié orthotrope quasi-homogène : deux modules anisotropes, à savoir, R_{0K}^{A*} , R_1^{A*} et l'angle polaire Φ_1^{A*} . Néanmoins, seuls les deux premiers paramètres sont considérés comme variables de conception pour des raisons décrites en détail dans [129]. En outre, pour paramétriser efficacement ce modèle, deux coefficients sont introduits, exprimés par :

$$\rho_0 = \frac{R_{0K}^{A*}}{R_0}, \quad \rho_1 = \frac{R_1^{A*}}{R_1}, \quad (4.12)$$

où R_0 et R_1 sont les modules anisotropes de la matrice de rigidité réduite du pli [93]. En outre, le changement de variable suivant permet d'éviter d'introduire les conditions de faisabilité sur ρ_0 et ρ_1 :

$$(\alpha_0, \alpha_1) := \left(\frac{\rho_0 - 1}{2(\rho_0^2 - 1)}, \rho_1 \right). \quad (4.13)$$

Différentes valeurs de (α_0, α_1) sont utilisées pour le panneau et le raidisseur, respectivement $(\alpha_{0_S}, \alpha_{1_S})$ et $(\alpha_{0_B}, \alpha_{1_B})$. Pour plus de détails sur la mise en œuvre du formalisme polaire dans l'analyse par EF des structures composites, se référer à [93, 124].

La configuration de référence du panneau raidi mono-raisseur est caractérisée par une valeur de la première charge de flambement égale à 445074 N [129] obtenue avec les valeurs de référence du Tab. 4.17. Une description détaillée de la stratégie numérique visant à récupérer les séquences d'empilement à partir de la définition des paramètres polaires du stratifié est proposée dans [129].

Le modèle EF non linéaire du panneau raidi, représenté dans la Fig. 4.22, est créé via des scripts ANSYS/APDL et son maillage est constitué d'éléments SHELL181 (éléments à quatre nœuds avec six degrés de liberté à chaque nœud). Un soin particulier a été apporté à la définition de la taille du maillage afin d'obtenir au moins trois éléments le long de la largeur de chaque panneau horizontal et de la partition des raidisseurs (segments de longueur a_1 , a_2 et a_3) et au moins six éléments pour les segments de hauteur h . Pour le panneau raidi de référence, le modèle EF consiste en un total de 4438 éléments. Pour imposer des conditions aux limites périodiques entre les bords opposés du panneau, des équations de contrainte appropriées ont été utilisées (voir [129] pour plus de détails sur ce point). Des éléments *Multi-Point Constraint* (MPC184) ont été créés et reliés à deux nœuds de référence des deux côtés : à $x = 0$, le nœud de référence concerné est encastré tandis que, à $x = b$, tous les degrés de liberté ont été mis à zéro, à l'exception du déplacement le long de l'axe x . Une force unitaire le long de l'axe x est appliquée sur ce nœud. Tous les détails concernant le modèle EF du panneau raidi sont disponibles dans [129].

La base de données d'entraînement \mathbf{Q}_t est générée via l'analyse EF de ce modèle en considérant $N = 6$ variables d'entrée et $M = 5$ sorties. Les variables d'entrée correspondent aux nombres de plis du panneau et du raidisseur, respectivement n_S et n_B , et aux paramètres matériau α_{0S} , α_{1S} , α_{0B} et α_{1B} . Les paramètres géométriques c_1 , c_2 et c_3 sont fixés à leur valeur de référence renseignées dans le Tab. 4.17. Quant aux sorties, elles comprennent :

- la charge critique λ_g , issue de l'analyse de flambement aux valeurs propres ;
- la valeur de la charge au point limite, ou *limit point*, noté λ_{lim} , correspondant à la charge maximale supportée avant effondrement de la structure. S'il y a un comportement en post-flambement, on considère la dernière valeur de la charge lorsque l'analyse de post-flambement s'arrête ; l'analyse en post-flambement est réalisée en utilisant le solveur *arc-length* [7] ;
- la valeur de la déformation axiale au centre de la peau ε_{x_S} , suivant l'axe x , lorsque la charge appliquée est égale à λ_g ;
- la valeur de la déformation axiale au centre de la flange supérieure du raidisseur ε_{x_B} , suivant l'axe x , lorsque la charge appliquée est égale à λ_g ;
- la raideur du panneau K_{panneau} , calculée à partir de la courbe force-déplacement de l'analyse non-linéaire.

4.3.3.2 Caractéristiques et solution du métamodèle

Le métamodèle employé pour approximer le comportement en flambement du motif élémentaire de ce panneau raidi est constitué de $N = 6$ variables d'entrée et de $M = 5$ sorties. Les caractéristiques de ce métamodèle sont renseignées dans la Tab. 4.15, tout

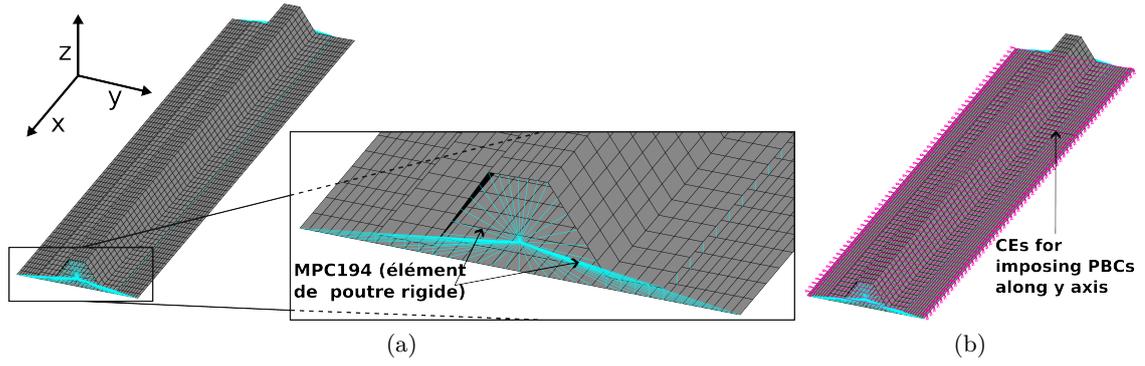


FIGURE 4.22 – (a) Modèle aux Éléments Finis de l’unité répétée du panneau raidi avec détails des éléments MPC184 [?] et (b) les contraintes pour imposer les conditions limites périodiques le long de l’axe y .

TABLE 4.15 – Caractéristiques du métamodèle pour reproduire le comportement du panneau raidi soumis à un chargement axial.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
n_S	$\mathcal{U}[20, 32] \in \mathbb{N}$							
n_B	$\mathcal{U}[20, 32] \in \mathbb{N}$							
α_{0_S}	$\mathcal{U}[0, 1]$	2	5	4	46656	15625	$]0, 1[$	$[1, 10]$
α_{1_S}	$\mathcal{U}[0, 1]$							
α_{0_B}	$\mathcal{U}[0, 1]$							
α_{1_B}	$\mathcal{U}[0, 1]$							

comme les intervalles de variation des variables d’entrée et des variables d’optimisation, i.e., β_i et ω_{i_1, \dots, i_N} . Sur un ordinateur Intel CORE i7 - 16 Go RAM, la base de données d’entraînement prend plus d’un mois à être construite dans la mesure où les simulations durent entre 60 s et 200 s. Ce temps a été réduit à une semaine en lançant plusieurs calculs simultanément. Néanmoins, pour limiter les coûts de calculs, la précision du métamodèle est évaluée à partir de simulations ponctuelles réparties sur l’ensemble des possibles plutôt qu’à partir d’une base de données de contrôle complète. Leur paramétrage est renseigné dans le Tab. 4.17. En outre, compte tenu que $r_k + 1 = 6$, le degré des polynômes de Bernstein sont fixés à $s_k = 2$ et le rapport τ déterminé par $n_k = r_k - 1$.

Dans ce cas là, l’indicateur de précision utilisé est la valeur moyenne des erreurs relatives évaluées pour toutes les sorties α exprimée par :

$$\bar{\varepsilon} = \frac{1}{M} \sum_{\alpha=1}^M \varepsilon_{\alpha}, \text{ avec } \varepsilon_{\alpha} = \frac{|q_{\alpha} - h_{\alpha}(\zeta_1, \dots, \zeta_N)|}{|q_{\alpha}|}. \quad (4.14)$$

Comme le montre le Tab. 4.16, la valeur de la fonction de coût a diminué de 35 % par rapport à sa valeur initiale, dont 27 % seulement grâce à l’optimisation des valeurs non triviales des vecteurs nodaux. Les résultats des simulations en vis-à-vis de leur équivalent calculé via le métamodèle sont regroupés dans le Tab. 4.18 avec l’erreur relative liée. On peut constater que les erreurs moyennes relatives entre le métamodèle et les résultats cal-

TABLE 4.16 – Évolution de f_{obj} en fonction de l'étape du processus d'optimisation.

f_{obj}		
$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$
1	0.73	0.65

culés par le modèle aux EF ne dépassent pas les 6 %. On peut conclure que le métamodèle résultant de cette étude est viable et peut être utilisé pour remplacer le modèle initial. En outre, l'emploi du métamodèle permet un réel gain de temps : la simulation aux EF met entre 60 s et 200 s à être menée tandis que le métamodèle met moins de 1 s pour fournir des résultats équivalents. Cet aspect peut être particulièrement intéressant dans une optique d'optimisation du modèle physique lui-même, demandeur en temps de calcul, ou pour mener à bien une Analyse de Sensibilité Globale.

TABLE 4.17 – Caractéristiques des bases de données d'entraînement et de contrôle utilisées pour l'étude de convergence.

	n_S	n_B	α_{0_S}	α_{0_B}	α_{1_S}	α_{1_B}
Simulation Réf.	28	32	0.079	0.044	0.57	0.50
Simulation 1	21	31	0.2	0.4	0.6	0.8
Simulation 2	23	29	0.4	0.6	0.8	0.2
Simulation 3	25	27	0.6	0.8	0.2	0.4
Simulation 4	27	25	0.8	0.2	0.4	0.6
Simulation 5	29	23	0.8	0.6	0.4	0.2
Simulation 6	31	21	0.2	0.6	0.4	0.8

TABLE 4.18 – Erreurs relatives moyennes entre les résultats fournies par les simulations de contrôle et le métamodèle.

Réf.	\mathbf{Q}_c						Métamodèle						$\bar{\varepsilon}$ [%]
	λ_g	λ_{lim}	ε_{x_S}	ε_{x_B}	E_S	E_B	λ_g	λ_{lim}	ε_{x_S}	ε_{x_B}	E_S	E_B	
1	4.65×10^5	6.97×10^5	-5.20×10^{-3}	-5.21×10^{-3}	1.49×10^5	1.49×10^5	4.54×10^5	6.93×10^5	-5.06×10^{-3}	-5.07×10^{-3}	1.50×10^5	1.50×10^5	1.74
2	2.50×10^5	3.74×10^5	-2.48×10^{-3}	-2.48×10^{-3}	1.68×10^5	1.68×10^5	2.43×10^5	3.69×10^5	-2.44×10^{-3}	-2.44×10^{-3}	1.67×10^6	1.67×10^6	1.58
3	1.90×10^5	2.85×10^5	-2.55×10^{-3}	-2.57×10^{-3}	1.24×10^5	1.24×10^5	1.77×10^5	2.66×10^5	-2.42×10^{-3}	-2.44×10^{-3}	1.22×10^6	1.22×10^6	4.86
4	2.24×10^5	3.36×10^5	-2.12×10^{-3}	-2.12×10^{-3}	1.76×10^5	1.76×10^5	2.08×10^5	3.14×10^5	-2.02×10^{-3}	-2.02×10^{-3}	1.72×10^6	1.72×10^6	5.11
5	3.24×10^5	4.86×10^5	-4.14×10^{-3}	-4.14×10^{-3}	1.30×10^5	1.30×10^5	3.08×10^5	4.47×10^5	-4.02×10^{-3}	-4.03×10^{-3}	1.27×10^6	1.27×10^6	4.20
6	2.84×10^5	4.03×10^5	-3.24×10^{-3}	-3.24×10^{-3}	1.46×10^5	1.46×10^5	2.70×10^5	3.93×10^5	-3.12×10^{-3}	-3.12×10^{-3}	1.44×10^6	1.44×10^6	3.34
	3.31×10^5	4.96×10^5	-2.90×10^{-3}	-2.90×10^{-3}	1.90×10^5	1.90×10^5	3.18×10^5	4.65×10^5	-2.82×10^{-3}	-2.82×10^{-3}	1.88×10^6	1.88×10^6	3.28

4.4 Conclusion du chapitre

Ce chapitre regroupe des cas d'applications avec différentes options spécifiques qui ont mis en avant les avantages et inconvénients d'utiliser un métamodèle basé sur une hypersurface NURBS développé dans ce travail. Les principales observations à retenir sont résumées dans le Tab. 4.19 et la liste ci-dessous :

- le métamodèle parvient à capturer de fortes non-linéarités, voire singularités géométriques, grâce à l'optimisation des valeurs non-triviales des vecteurs nodaux. Néanmoins, plus le problème étudié sera non-linéaire, plus la quantité de données à considérer pour le construire sera importante. Dans certaines situations, cela peut même être contre-productif si elles sont trop coûteuses à obtenir ;
- si les zones non-linéaires sont déjà capturées et que l'optimisation des valeurs non-triviales des vecteurs nodaux est inefficace, l'optimisation des poids associés aux CPs permet tout de même de modifier la courbure du métamodèle pour qu'il se rapproche au plus juste des TPs ;
- un terme de lissage peut être intégré dans le processus de génération du métamodèle pour non seulement réduire les éventuelles non-linéarités parasites, tel que du bruit, mais également pour limiter le phénomène d'*overfitting* et ainsi pouvoir prendre un rapport τ plus proche de l'unité, nécessaire dans des situations où les données d'entraînement sont limitées ;
- le métamodèle est très rapide à l'emploi. Une fois construit, il fournit des résultats en un temps inférieur à la seconde, quelque soit le problème d'origine et le nombre de ses dimensions d'entrée et de sortie. Néanmoins, le processus d'optimisation sous contraintes, peut durer de quelques secondes (problème de Poisson) à quelques heures selon le modèle (problème du panneau raidis) ;
- le métamodèle employé est construit à partir d'une base de données elle-même générée à partir de résultats répartis dans l'espace selon une grille uniforme. Cette particularité est la limite principale de cette méthode : en montant en dimensions d'entrée, le coût d'évaluation des bases de données croît exponentiellement et la question sur l'utilité d'employer un métamodèle peut se poser (exemple du panneau raidis). Une réflexion est donc à avoir sur l'emploi d'une méthode d'échantillonnage plus efficace pour générer les bases de données et sur la formulation vectorielle du problème d'optimisation, et non matricielle ;

La technique de métamodélisation basée sur une hypersurface NURBS est bien adaptée à des cycles de conception type *V-and-V*, et plus particulièrement à la méthode MHMN développée dans ce travail de recherche. Le faible coût de sa partie *on-line* va lui permettre de communiquer en temps réel lors de la résolution du modèle global MBSE.

TABLE 4.19 – Récapitulatif des avantages et inconvénients du métamodèle basé sur une hypersurface NURBS pour les différents cas d’application vu dans ce chapitre.

Cas test	Caractéristique du problème		Métamodèle		
	Non-linéarité ¹	Complexité ²	Données requises ³	Précision ⁴	Temps <i>off-line</i> ⁵
Poisson	--	--	--	++	seconde
Amplitude charriot	-	--	--	++	seconde
Déplacement charriot	+	-	+	++	minute
Fonction de Salomon	++	+	++	-	seconde
Plaque pleine	-	+	+	++	jour
Plaque percée	+	++	+	+	jour
Panneau raidi	+	++	+	+	semaine

¹ Pic, oscillations, variations haute fréquence, singularités.

² Temps de calcul, physiques en jeu, mise en place du modèle.

³ Valeurs des indicateurs R_α^2 , $\bar{\varepsilon}_\alpha$, \mathbf{e}_{M_α} , f_{obj} .

⁴ Taille de la base de données d’entraînement.

⁵ Comprend le temps de construction des bases de données d’entraînement et de contrôle ainsi que la mise en place de tout le processus de génération du métamodèle, calcul d’erreur compris.

5

Application de la stratégie de modélisation multifidélité à un problème thermomécanique

Sommaire

5.1	Introduction au chapitre	153
5.2	Première modélisation MBSE et définition des objectifs	154
5.2.1	Analyse MBSE du Système Embarqué à Température Contrôlée-1	154
5.2.1.1	Définition et exigences MBSE	154
5.2.1.2	Modèle MBSE et contrôle des exigences	157
5.2.2	Analyse MBSE Système Embarqué à Température Contrôlée-2	158
5.2.2.1	Rôle et enjeu du Matériau à Changement de Phase	158
5.2.2.2	Modèle MBSE	158
5.2.2.3	Résultats, limites et contraintes	160
5.3	Application de l'ASG pour la mise en évidence des sous-systèmes influents	161
5.3.1	Formulation du problème	162
5.3.2	Résultats numériques	163
5.4	Génération du métamodèle basé sur les hypersurfaces NURBS	164
5.4.1	Zones critiques et modèle EF	164
5.4.2	Génération du métamodèle comme solution d'un problème d'optimisation	167
5.4.3	Précision du métamodèle	169
5.5	Stratégie hybride haute-fidélité	171
5.5.1	Construction du modèle hybride haute-fidélité du SETC-3	171
5.5.2	Initialisation du modèle hybride haute-fidélité	171
5.5.3	Résultats du modèle hybride haute-fidélité SETC-3	172
5.6	Conclusion	175

RÉSUMÉ DU CHAPITRE

Ce chapitre propose l'application de la stratégie de modélisation hybride multifidélité sur un système embarqué mettant en jeu des phénomènes couplés de nature thermomécanique. Outre l'application de notre stratégie, les objectifs sont l'étude du comportement mécanique du système en réponse à des sollicitations thermomécaniques et l'identification des solutions permettant de limiter la température et les contraintes des régions critiques. Le cas d'étude se compose d'un empilement de plusieurs matériaux et composants englobés dans une enveloppe isolante. Les étages thermiques d'intérêts sont une carte électronique d'acquisition de données et une carte électronique de calcul et de transmission de données. Lors de son fonctionnement, cette dernière dissipe un flux de chaleur important dans l'ensemble du dispositif, ce qui fait croître rapidement la température des cartes électroniques. Le schéma fonctionnel de ce système nommé Système Embarqué à Température Contrôlée (SETC) est représenté dans la Fig. 5.1. Un Matériau à Changement de Phase (MCP) est positionné sous la carte électronique de puissance pour absorber une partie de l'énergie dissipée et limiter ainsi l'augmentation de la température. Cependant, non seulement la température du MCP va évoluer rapidement, mais sa géométrie va également se déformer par dilatation et entraîner des efforts mécaniques sur l'enveloppe, jusqu'à sa ruine dans certaines situations. Cet étage thermique est donc considéré par les physiciens et le bureau d'études, comme influent et critique, ce qui sera confirmé par les résultats d'une Analyse de Sensibilité Globale (ASG). Sa description par approche Model-Based System Engineering (MBSE) est alors raffinée par un métamodèle basé sur une hypersurface Non-Uniform Rational Basis-Spline (NURBS), entraîné sur des résultats haute-fidélités.

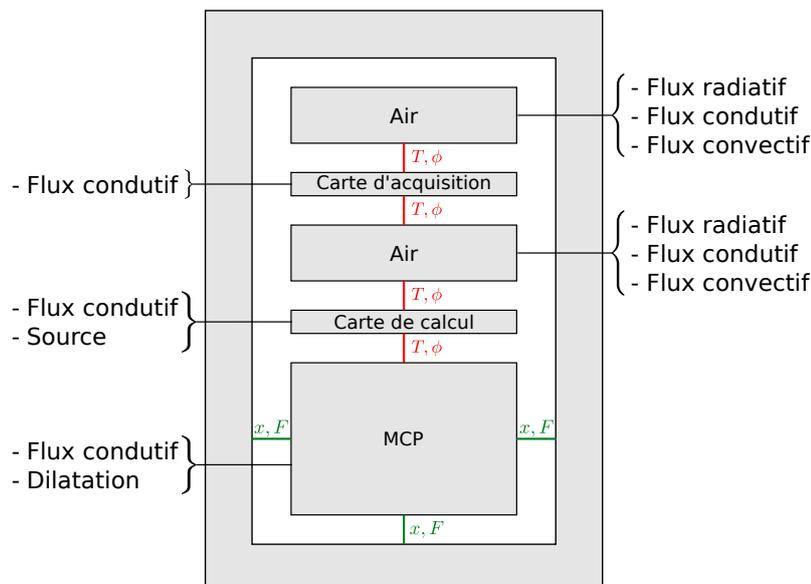


FIGURE 5.1 – Schéma fonctionnel du Système Embarqué à Température Contrôlée. Le schéma identifie les principales couches thermiques, leur comportement thermique et les variables de communication.

5.1 Introduction au chapitre

La stratégie MHMN développée dans cette thèse permet de décrire un dispositif en le séparant en plusieurs sous-systèmes de façon adaptative, en fonction des capacités de modélisation et des attendus en précision. Selon le besoin, ces sous-systèmes peuvent être décrits par un système analytique d'EAD ou par un métamodèle. Ce dernier va être entraîné par des calculs haute-fidélité ou par des données issues d'expériences. Pour mettre en place cette stratégie, trois étapes principales sont requises.

La première consiste à réaliser un premier modèle du système d'étude via une approche MBSE. Le but de cette étape est de décomposer le dispositif d'étude en plusieurs sous-systèmes, faisant appel à différentes physiques, reliés par des échanges de flux et de potentiels. Ces sous-systèmes sont décrits par des systèmes d'EAD pré-conçus ou développés par l'utilisateur. Néanmoins, comme évoqué dans le chapitre 1, ce type d'approche comporte des limites : elle est généralement destinée à réaliser des modèles dynamiques avec peu de degrés de liberté (0-D, voire 1-D) et les systèmes d'EAD comportent souvent des hypothèses simplificatrices.

La seconde étape consiste à déterminer les zones où une approche analytique est insuffisante. Cela peut être en raison de la présence de gradients des variables d'état dans la région considérée, du couplage de phénomènes physiques rendant impossible une simplification système, ou encore de la difficulté à identifier les points de raccordement avec les autres sous modèles. En général, ces zones sont identifiées par l'avis critique du concepteur ou de l'architecte produit qui peut être appuyé par une ASG. C'est l'approche que nous privilégions dans ce travail. Par la suite, ces zones seront appelées zones critiques. Le terme influent sera employé pour désigner les zones dont le comportement influence fortement la sortie observée. Les ASG utilisées ici sont quantitatives : elles permettent de quantifier l'influence des paramètres de conception du système d'étude par rapport à la sortie considérée. Ces analyses sont directement appliquées sur le modèle MBSE, exporté au format FMU, pour assurer une analyse rapide et à moindre coût de calcul.

Les simulations MBSE sont réalisées avec le logiciel Dymola inclus dans la plateforme 3D Experience de Dassault System¹. Rappelons que Dymola fait appel au langage de programmation Modelica dont la grammaire est assez proche du langage C/C++ ce qui le rend compatible de toutes les technologies modernes d'échange de données (fichier FMU, gestionnaire de librairie CMake).

Pour finir, la stratégie MHMN de cette thèse, et ce qui constitue l'originalité de ce travail, permet d'inclure un métamodèle haute-fidélité dans le modèle global MBSE, à la place des sous-systèmes critiques et/ou influents identifiés lors de la deuxième phase. Le métamodèle en question se base sur l'emploi et l'optimisation d'hypersurfaces NURBS, construites à partir d'une base de données haute-fidélité (voir chapitres 3 et 4) et interfacées

1. <https://www.3ds.com/fr/>

avec le modèle MBSE via le gestionnaire de librairie CMake.

Ce chapitre propose de mettre en exergue la stratégie MHMN sur un système embarqué thermomécanique nommé Système Embarqué à Température Contrôlée (SETC), pouvant être employé dans l'industrie aéronautique ou dans l'aérospatial. Il s'agit d'un cas d'école simplifié qui contient toutes les difficultés de modélisation qui peuvent être rencontrées sur des problèmes concrets qui limitent l'application d'une approche MBSE classique. L'objectif est alors d'appliquer la MHMN pour combiner la facilité d'utilisation d'une approche MBSE à des calculs haute-fidélités. Le problème thermomécanique se prête bien à cet exercice dans la mesure où la modélisation thermique peut se faire via une approche MBSE classique mais que le couplage thermomécanique, qui va permettre d'obtenir une précision adaptée à la finesse de résolution, requiert une attention particulière.

Dans ce chapitre, la section 5.2 présente le dispositif étudié, ses exigences et l'emploi de l'approche MBSE classique dans sa calibration. Les limites de cette approche sont ensuite définies. La section 5.3 permet de confirmer l'intérêt d'employer l'approche MHMN via une ASG. La section 5.4 est consacrée à la génération du métamodèle haute-fidélité dont le but est de combler les limites du modèle initial MBSE. La section 5.5 présente alors le modèle MHMN du système embarqué. Pour finir, la section 5.6 apporte la conclusion et des perspectives à ce travail.

5.2 Première modélisation MBSE et définition des objectifs

Comme illustré dans la Fig. 5.2, la mise en place complète de la stratégie de modélisation hybride haute-fidélité se déroule en quatre étapes. La première, présentée dans cette section, consiste à définir le dispositif étudié, ses limites et ses besoins de modélisation.



FIGURE 5.2 – Première étape de la stratégie Modélisation Hybride MBSE - NURBS : l'analyse *Model-Based System Engineering*.

5.2.1 Analyse MBSE du Système Embarqué à Température Contrôlée-

1

5.2.1.1 Définition et exigences MBSE

Ce chapitre propose l'étude d'un système embarqué simple, de mission de courte durée : $t_{\text{tot}} = 100$ s. Le système est axisymétrique : il est compact tout en assurant la collecte d'information, son analyse puis sa transmission vers l'utilisateur. Pour parvenir à cela, il

se compose d'une première carte électronique d'acquisition qui contient un capteur à faible dissipation thermique, et d'une seconde carte électronique de calcul assurant l'analyse et la transmission des données. Lors du fonctionnement de la seconde carte, une puissance thermique importante est dissipée dans l'ensemble du système. Le rôle des autres composants est de protéger la carte électronique d'acquisition de cette énergie. Il s'agit de la première exigence à respecter : le seuil thermique de fonctionnement de la carte d'acquisition est fixé à $T_{\max, \text{Acier}} = 373 \text{ K}$.

Pour protéger l'ensemble du dispositif des agressions extérieures, tant thermiques que mécaniques, une enveloppe en mullite est ajoutée. La seconde exigence de fonctionnement à respecter est d'assurer la tenue thermomécanique de l'enveloppe face aux contraintes et déformations internes du dispositif. La première conception du SETC est représentée par le schéma de la Fig. 5.3 avec de haut en bas :

1. une couche d'air isolant la carte électronique d'acquisition de l'enveloppe ;
2. la carte électronique d'acquisition, fixée à une plaque support. Cette dernière, faite en acier, est considérée comme infiniment rigide. L'épaisseur de la carte électronique étant négligeable par rapport à celle de son support, son comportement thermique sera identique à celui du support ;
3. une couche d'air isolant les deux cartes électroniques ;
4. la carte électronique de calcul, fixée à une plaque support. Cette dernière, en cuivre, est également considérée comme infiniment rigide. L'épaisseur de la carte électronique étant négligeable devant celle de son support, son comportement thermique sera identique à ce dernier ;
5. une enveloppe en mullite qui contient l'ensemble du système embarqué.

Les couches thermiques sont régies par les lois de comportement thermiques classiques renseignées dans la Fig. 5.1. Un premier modèle MBSE, nommé SETC-1, est réalisé pour étudier les comportements de chaque sous-système. Les hypothèses suivantes sont considérées pour sa mise en place :

- les couches thermiques ont un comportement homogène, justifié par un nombre de Biot petit devant l'unité [95] ;
- une découpe en sous-systèmes analytiques est possible ;
- il s'agit d'un problème purement thermique au premier abord, avec un comportement temporel dynamique, piloté par la production d'énergie interne.

Le modèle MBSE résultant, équivalent à un problème d'ordre 0, est présenté dans la Fig 5.4. On peut y retrouver les sous-systèmes inhérents à chaque couche thermique. Chacun d'eux est construit par assemblage de blocs physiques élémentaires comme illustré pour la plaque d'acier, qui communiquent via l'échange d'une température $T(t)$ et d'un flux de chaleur $\phi(t)$. Le détail de la construction analytique de ces blocs élémentaires est

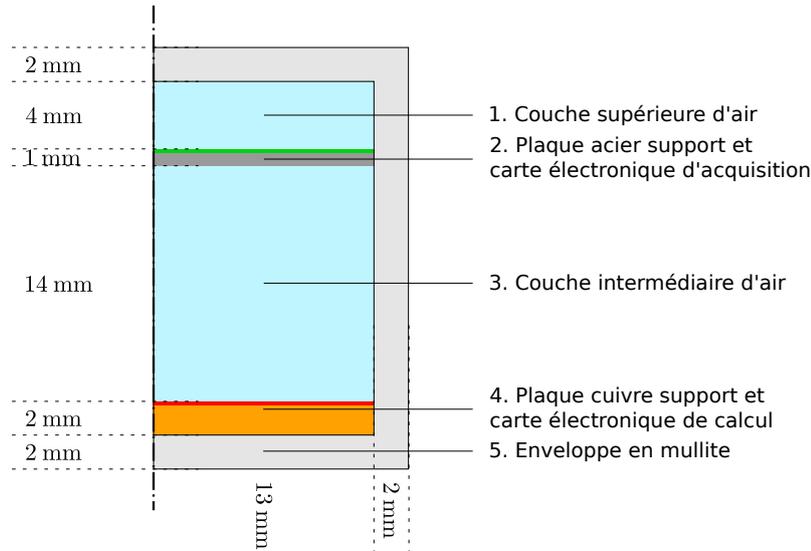


FIGURE 5.3 – Schéma du Système Embarqué à Température Contrôlée en coupe avant suivant son axe de symétrie axial. Il se compose, dans sa partie inférieure, d'une plaque supportant la carte électronique de calcul et, dans sa partie supérieure, d'une carte électronique d'acquisition. Les autres couches thermiques correspondent à des couches isolantes d'air. L'ensemble est entouré d'une enveloppe en mullite.

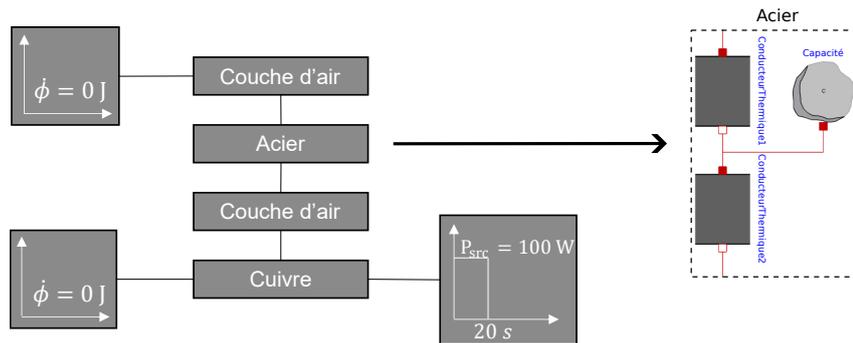


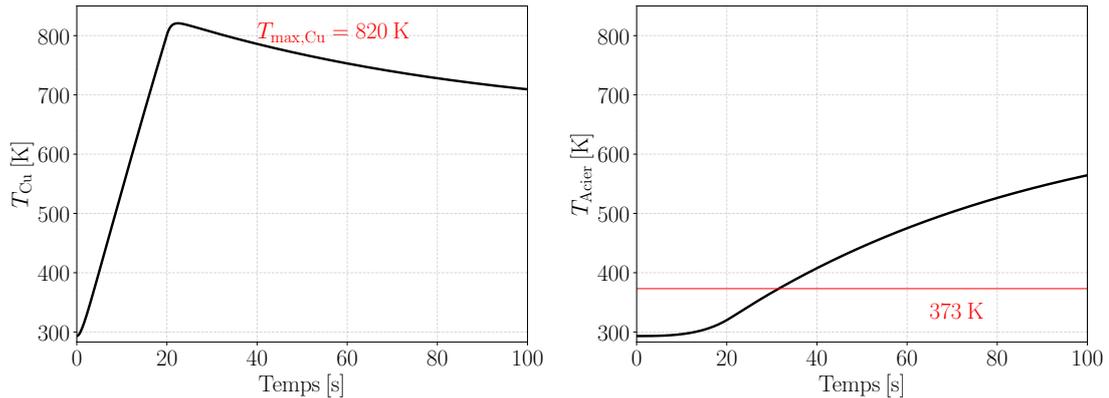
FIGURE 5.4 – A gauche, modèle *Model-Based System Engineering* du système initial SETC-1. A droite, détail du sous-système conductif relatif à la plaque d'acier.

donné en annexe E.1 et celui des sous-systèmes globaux par l'annexe E.2. On remarquera que la mullite est modélisée par des flux thermiques nuls, jouant le rôle de conditions aux limites adiabatiques, dans la mesure où il s'agit d'un matériau isolant. Pour finir, le sous-système de la source de flux est lié à l'étage de la carte électronique de calcul. Cette source correspond à l'application d'une puissance de $P_{src} = 100 \text{ W}$ pendant une durée de $\Delta t_{calc} = 20 \text{ s}$.

La section suivante présente les caractéristiques thermiques de ce premier modèle ainsi que ses résultats. Dans la suite de ce chapitre, ce cas d'étude n'a pas de réalité industrielle ni de prototype de laboratoires, les différentes grandeurs thermique, mécanique et géométrique employées sont tirées de la littérature et de réflexions internes avec le bureau d'étude du CEA.

TABLE 5.1 – Caractéristiques matériaux du modèle *Model-Based System Engineering* du SETC-1.

Grandeur	Symbole	Cuivre	Acier	Air
Capacité massique [$\text{J kg}^{-1} \text{K}^{-1}$]	c_p	383	435	1
Conductivité [$\text{W m}^{-1} \text{K}^{-1}$]	λ	386	54	0.026
Émissivité [-]	ϵ	0.7	0.8	-
Masse volumique [kg m^{-3}]	ρ	8954	7700	1004



(a) Évolution de la température de la plaque de cuivre au cours du temps du modèle SETC-1. (b) Évolution de la température de la plaque d'acier au cours du temps du modèle SETC-1.

FIGURE 5.5 – Évolution des températures des éléments d'intérêt du système embarqué initial, modélisé par approche MBSE.

5.2.1.2 Modèle MBSE et contrôle des exigences

Une fois le modèle MBSE mis en place, il est paramétré selon les grandeurs renseignées dans le Tab. 5.1. Quant à sa géométrie, dont les dimensions sont renseignées dans la Fig. 5.3, elle est choisie pour correspondre aux dimensions pratiquées dans la conception de systèmes embarqués. Les simulations sont réalisées pour la durée totale de la mission, à savoir $t_{\text{tot}} = 100$ s. Les évolutions des températures de la plaque en cuivre et de la plaque en acier sont respectivement tracées par les graphiques dans les Figs. 5.5a et 5.5b.

L'évolution de la température de la carte électronique de calcul est marquée par un pic aux alentours de $t = 20$ s, temps au bout duquel son activité s'arrête. La valeur maximale atteinte est de $T_{\text{max,Cu}} = 820$ K, inférieure à la température de fusion du cuivre $T_{\text{Cu,fus}} = 1356$ K : cette couche n'est pas détériorée. L'énergie est ensuite dissipée dans le dispositif, et en particulier vers la carte électronique d'acquisition à protéger. Sa température $T_{\text{max,Acier}} = 560$ K dépasse la température de fonctionnement admissible $T_{\text{max,adm}} = 373$ K en seulement $\Delta t = 36$ s. La conclusion est immédiate : la carte électronique supérieure ne vérifie pas les exigences et la mission ne pourra pas être menée à son terme.

Pour pouvoir remplir les exigences de fonctionnement du produit, la solution proposée

est d'employer une pile thermique sous la forme d'un MCP, fixée sous la source de chaleur. L'idée est que cette pile thermique absorbe l'énergie dissipée par la source pour limiter le flux thermique vers la carte électronique d'acquisition.

5.2.2 Analyse MBSE Système Embarqué à Température Contrôlée-2

5.2.2.1 Rôle et enjeu du Matériau à Changement de Phase

L'emploi des MCP s'est principalement développé dans le domaine du stockage de l'énergie. Leur forte chaleur latente couplée à leur capacité thermique font qu'une importante quantité de chaleur est requise pour opérer un changement d'état. Ce phénomène est particulièrement utile pour réaliser du stockage d'énergie [54] : de l'énergie est fournie au MCP pour qu'il change de phase, puis elle est restituée par le processus inverse. Des applications de ce phénomène peuvent se retrouver dans l'exploitation de l'énergie solaire [86, 168, 184].

Dans le but d'améliorer la conductivité de ces MCP, et donc l'homogénéité du dispositif, plusieurs méthodes ont vu le jour [25]. Les principales sont l'encapsulation microscopique [55, 184] ou macroscopique [86] et l'emploi de matrices conductrices [169]. Les propriétés thermiques du matériau résultant sont similaires dans les deux situations. Les matrices dopantes employées peuvent être des mousses en cuivre [169] ou encore un matériau poreux [194].

L'emploi d'un MCP introduit une nouvelle problématique : la prise en compte et la modélisation de leur dilatation. Cette dilatation varie d'un type de MCP à un autre mais reste importante, de l'ordre 10 % de la longueur caractéristique. Lors de la conception des supports de ces matériaux, elle doit donc être prise en compte pour s'assurer de leur résistance aux efforts de pression.

En considérant cette nouvelle situation, un second modèle MBSE a été construit pour contrôler le respect des exigences de fonctionnement du système embarqué. La physique à considérer n'est plus simplement thermique mais thermomécanique.

5.2.2.2 Modèle MBSE

Le modèle MBSE de la seconde conception du SETC, nommé SETC-2, est également construit par assemblage de sous-systèmes, eux-mêmes construits à partir d'assemblage de blocs élémentaires physiques dont le détail est donné en annexe E.2 et E.3. Ce nouvel assemblage et ses dimensions sont illustrés dans la Fig. 5.6. On peut y retrouver les mêmes sous-systèmes que pour le modèle SETC-1, si ce n'est l'ajout de l'élément MCP et d'une rondelle isolante. Cette dernière est employée uniquement pour diriger le flux thermique vers le MCP et ne fera pas l'objet d'une attention particulière.

La particularité de ce nouveau modèle est la nécessité de développer un sous-système compatible d'une approche MBSE homogène visant à modéliser le comportement ther-

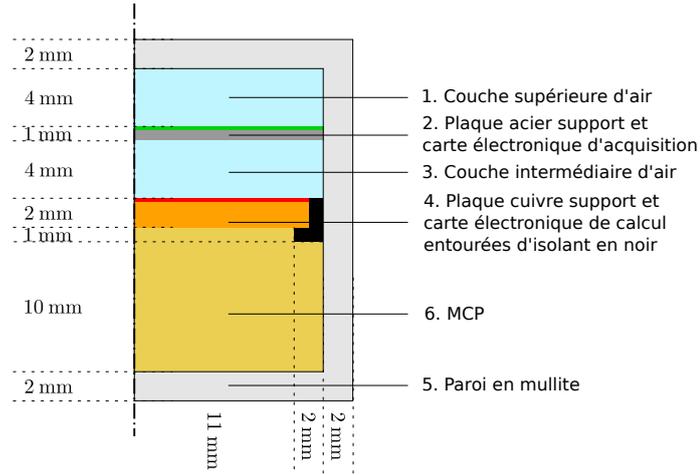


FIGURE 5.6 – Schéma du système embarqué SETC-2 comportant un Matériau à Changement de Phase. Un Matériau à Changement de Phase a été introduit.

momécannique de l'ensemble MCP - mullite. Le changement de phase est considéré comme homogène en volume, β , la portion de la phase liquide exprimée plus loin. Pour l'aspect thermique, seul le MCP est pris en compte, régi par [73] :

$$\phi_{\text{capa,MCP}} = mc_{p,\text{eq}} \frac{\partial T}{\partial t}, \quad (5.1)$$

avec $\phi_{\text{capa,MCP}}$ le flux capacitif du MCP et $c_{p,\text{eq}}$ la capacité calorifique massique à pression constante équivalente de ce matériau. Le comportement lors du changement de phase, et tout au long du cycle thermique, est contenu dans l'expression de $c_{p,\text{eq}}$:

$$c_{p,\text{eq}} = (1 - \beta)c_{p_S} + \beta c_{p_L} + \frac{\partial \beta}{\partial T} \Delta H_m, \quad (5.2)$$

avec β la fraction liquide du MCP, c_{p_S} et c_{p_L} respectivement ses capacités calorifiques à l'état solide et liquide et ΔH_m son enthalpie de changement de phase, appelée aussi chaleur latente de changement de phase. La fraction liquide, β , permet d'évaluer la quantité de liquide présente dans le volume de matière et d'adapter l'impact des coefficients c_{p_S} , c_{p_L} et ΔH_m . Cette fraction liquide s'exprime par :

$$\beta = \frac{1}{\pi} \left[\arctan \left(\frac{2\gamma(T - T_{\text{fus}})}{\Delta T} \right) + \frac{\pi}{2} \right], \quad (5.3)$$

où T_{fus} correspond à la température de changement de phase moyenne du MCP, ΔT sa plage de température de changement de phase et γ un coefficient de changement de phase, fixé à $\gamma = 3.1$ par [73]. L'importance des coefficients c_{p_S} et c_{p_L} est directement dépendante de la valeur de β . Sa chaleur latente ΔH_m quant à elle est exprimée par :

$$\frac{\partial \beta}{\partial T} = \left(\frac{2\gamma}{\Delta T} \right)^{-1} \left[\pi \left[\left((T - T_{\text{fus}}) \left(\frac{2\gamma}{\Delta T} \right) \right)^2 + 1 \right] \right]. \quad (5.4)$$

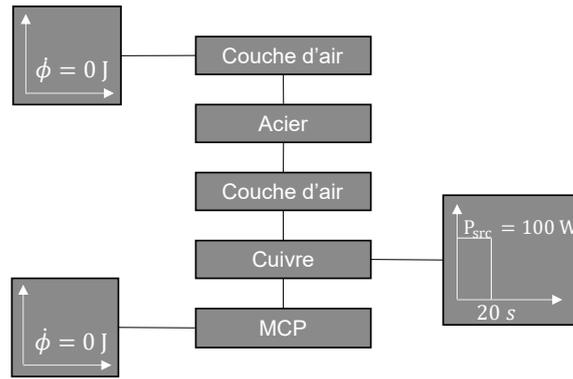


FIGURE 5.7 – Modèle *Model-Based System Engineering* du SETC-2 contenant le Matériau à Changement de Phase.

TABLE 5.2 – Caractéristiques thermiques du Matériau à Changement de Phase.

Grandeur	Symbole	MCP
Capacité thermique solide [$\text{J kg}^{-1} \text{K}^{-1}$]	c_p	2000
Conductivité solide [$\text{W m}^{-1} \text{K}^{-1}$]	λ_{MCP}	2.5
Masse volumique solide [kg m^{-3}]	ρ_S	880
Chaleur latente [J kg^{-3}]	ΔH_m	174000
Capacité thermique liquide [$\text{J kg}^{-1} \text{K}^{-1}$]	c_p	2000
Conductivité liquide [$\text{W m}^{-1} \text{K}^{-1}$]	λ_{MCP}	2.5
Masse volumique liquide [kg m^{-3}]	ρ_L	770

Quant au flux conductif du MCP, il est formulé par :

$$\phi_{\text{cond,MCP}} = \frac{\lambda_{\text{eq}} A}{e} \Delta T \quad (5.5)$$

avec $\phi_{\text{cond,MCP}}$ le flux échangé par le MCP et λ_{eq} sa conductivité équivalente. En considérant la phase liquide β , cette conductivité peut s'exprimer par :

$$\lambda_{\text{eq}} = (1 - \beta)\lambda_S + \beta\lambda_L, \quad (5.6)$$

avec λ_S la conductivité du matériau dans sa phase solide et λ_L dans sa phase liquide.

Pour finir, la contrainte mécanique subie par la mullite est déduite de la dilatation théorique du MCP le long de son rayon et de son épaisseur. Cet allongement est calculé en fonction du coefficient de dilatation du MCP fixé à $\alpha \approx 10^{-4} \text{K}^{-1}$. En comparaison, ce coefficient est en moyenne de $\alpha \approx 10^{-6} \text{K}^{-1}$ pour les métaux.

5.2.2.3 Résultats, limites et contraintes

Pour finir de définir le modèle MBSE, les paramètres thermiques du MCP, tirés de la littérature [55,86,169,184] et du fabricant RUBITHERM² sont renseignés dans le Tab. 5.2. La géométrie est définie dans la Fig. 5.6.

2. <https://www.rubitherm.eu/en/>

La Fig. 5.8a montre que l'évolution de T_{Cu} est marquée par un maximum aux alentours de $t = 20$ s d'une valeur de $T_{\text{max,Cu}} = 600$ K, plus faible que dans le SETC-1 alors de $T_{\text{max,Cu}} = 820$ K. Dans cette situation également le cuivre support de la carte de calcul n'est pas détérioré. L'énergie est dissipée ensuite dans le dispositif jusqu'à atteindre la plaque en acier. La température de cette dernière ne dépasse pas $T_{\text{max,Acier}} < 330$ K, comme illustré dans la Fig. 5.8b. La première exigence thermique est respectée.

En parallèle de cette étude thermique, les calculs d'ordre de grandeur de la dilatation du MCP sont effectués. Les longueurs caractéristiques finales du MCP sont de : $\Delta r_{\text{MCP}} = 0.56 \times 10^{-3}$ m et $\Delta e_{\text{MCP}} = 0.447 \times 10^{-3}$ m. En considérant les ordres de grandeurs de ces sorties, la pression équivalente exercée sur les parois en mullite est estimée à :

$$P = \frac{F}{S} \approx \frac{k\Delta r_{\text{MCP}}}{S} \approx \frac{10^8 \times 10^{-3}}{10^{-4}} \approx 10^7 \text{ Pa}, \quad (5.7)$$

avec k la raideur équivalente de la mullite. La contrainte subie par la mullite est du même ordre de grandeur que sa limite à la rupture [58].

Le dimensionnement du dispositif SETC-1 a permis de trouver les tendances de comportement des différentes entités physiques qui le composent. Le non respect des exigences de fonctionnement thermique a notamment été mis en évidence. La conception du SETC-2 a permis de proposer une solution à ce problème. Néanmoins nous sommes proches des limites de fonctionnement et ces résultats étant approximatifs, ils ne permettent pas de conclure avec certitude sur la viabilité de ce nouveau dispositif pour la seconde exigence mécanique. Une étude plus approfondie de la contrainte subie par la mullite via une analyse aux EF est donc nécessaire.

La mise en place de la stratégie MHMN va permettre d'apporter de la précision au modèle global MBSE tout en conservant son efficacité d'utilisation. L'objectif est alors d'entraîner un métamodèle basé sur une hypersurface NURBS par des résultats issus d'un modèle EF haute-fidélité. Néanmoins, une ASG est réalisée auparavant sur le SETC-2 dans le but de confirmer l'influence de la zone critique sur le comportement du SETC mais également de déterminer les paramètres de conception influents à prendre en compte comme variables d'entrée du métamodèle.

5.3 Application de l'ASG pour la mise en évidence des sous-systèmes influents

Comme illustré dans la Fig. 5.9, la mise en place complète de la stratégie de modélisation hybride haute-fidélité se déroule en quatre étapes. La seconde, réalisée dans cette section, consiste à identifier les sous-systèmes influents.

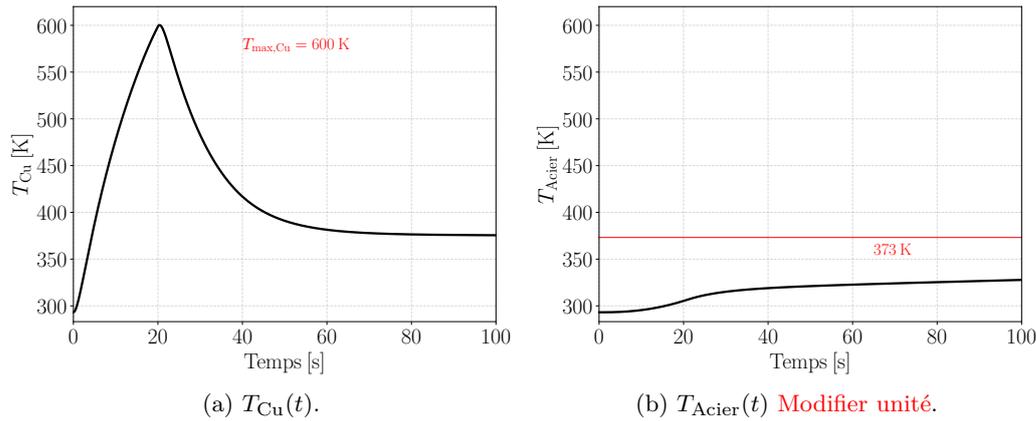


FIGURE 5.8 – Évolution des températures moyennes des plaques en cuivre, T_{Cu} , et en acier, T_{Acier} au cours du temps, ainsi que du rayon r_{MCP} et de l'épaisseur e_{MCP} du Matériau à Changement de Phase, s'il n'était pas contraint par l'enveloppe en mullite.



FIGURE 5.9 – Seconde étape de la stratégie Modélisation Hybride MBSE - NURBS : Analyse de Sensibilité Globale

5.3.1 Formulation du problème

L'objectif de cette ASG est de déterminer quels paramètres sont les plus influents sur la température maximale de la carte électronique d'acquisition et de savoir sur quelle partie du modèle il faut porter un effort de modélisation accru. La méthode ASG employée est celle basée sur les indices de Sobol, présentée au chapitre 2 ainsi que dans [88, 180].

Les différentes variables de conception retenues pour cette ASG, ainsi que leur valeur de référence et leur plage de variation uniforme, notée $\mathcal{U}[\cdot, \cdot]$, sont rassemblées dans le Tab. 5.3. On y retrouve la conductivité thermique du MCP λ_{MCP} , sa chaleur latente ΔH_m et sa température de fusion $T_{fus,MCP}$. L'émissivité des parois entourant la couche d'air intermédiaire, ϵ_{air} , est aussi considérée. Elle traduit l'importance du rayonnement thermique émis par les parois en cuivre et en acier vers l'air. Enfin, l'épaisseur de cette couche d'air, e_{air} , est également prise en compte. Les plages de variations de ces variables sont définies pour que le critère de fonctionnement thermique du dispositif soit respecté, i.e., $T_{max,adm} = 373$ K. Le but est d'étudier l'influence de ces variables sur la température finale atteinte par la plaque en acier $T_{max,acier}$, qui est aussi sa température maximale.

Pour que le calcul des indices de Sobol réalisé avec l'algorithme FAST ait une incertitude acceptable, un échantillonnage de 2^{12} simulations a été considéré, respectant ainsi le critère de convergence fixé au chapitre 2. Ces simulations se basent sur le modèle MBSE du SETC-2 (Fig. 5.7) pour limiter le temps de calcul de l'ASG.

TABLE 5.3 – Valeurs de référence et intervalles de variation.

Variable	Référence	Intervalle
λ_{MCP} [$\text{W m}^{-1} \text{K}^{-1}$]	2.5	$\mathcal{U}([2.0, 4.0])$
ΔH_m [J kg^{-1}]	174000	$\mathcal{U}([150000, 250000])$
$T_{\text{fus,MCP}}$ [K]	350	$\mathcal{U}([350, 363])$
ϵ_{air} [-]	0.6	$\mathcal{U}([0.5, 0.7])$
e_{air} [m]	0.004	$\mathcal{U}([0.002, 0.006])$

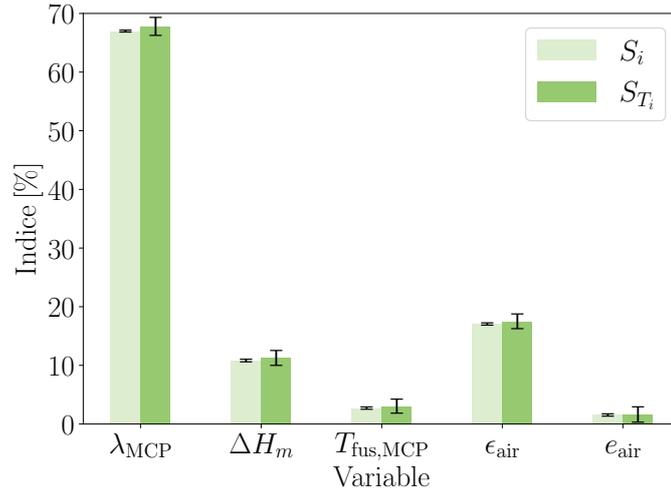


FIGURE 5.10 – Résultat de l'Analyse de Sensibilité Globale sur la température maximale atteinte par la carte électronique d'acquisition, dans le SETC-2.

5.3.2 Résultats numériques

Les résultats de l'ASG sont présentés dans l'histogramme de la Fig. 5.10 avec les incertitudes de mesure en noir. L'incertitude moyenne sur ces résultats étant de $\pm 7.53 \times 10^{-3}$ points pourcentages, on considère qu'ils sont exploitables. L'écart entre les indices élémentaires et les indices totaux étant inférieur à 1 %, on peut déduire que les interactions entre ces variables ont un rôle mineur sur la valeur finale de la température de la plaque d'acier.

La variable la plus influente sur $T_{\text{max,acier}}$ est la conductivité du MCP avec un indice élémentaire de $S_{\lambda_{\text{MCP}}} = 67$ % et total de $S_{T_{\lambda_{\text{MCP}}}} = 68$ %. Les deux autres variables ayant une influence non négligeable sont la chaleur latente de fusion du MCP et l'émissivité dans la cavité d'air intermédiaire avec des indices de Sobol compris entre 10 % et 20 %. Pour finir, la température de fusion du MCP n'est pas influente sur la sortie tout comme l'épaisseur de la cavité d'air intermédiaire avec des indices inférieurs à 4 %.

Au vu de ces résultats, le sous-système relatif au MCP peut-être qualifié de sous-système influent en plus d'être critique par son besoin en résolution spatiale (gradient de déplacement et de température dus au changement de phase). Cela motive notre intuition à utiliser des simulations haute-fidélité pour ce sous-système.

5.4 Génération du métamodèle basé sur les hypersurfaces NURBS

Comme illustré dans la Fig. 5.11, la mise en place complète de la stratégie de modélisation hybride haute-fidélité se déroule en quatre étapes. La troisième, réalisée dans cette section, consiste à générer le métamodèle basé sur une hypersurface NURBS.



FIGURE 5.11 – Troisième étape de la stratégie MHMN : génération du métamodèle basé sur une hypersurface NURBS.

5.4.1 Zones critiques et modèle EF

La partie inférieure du SETC-2 correspondant à l'ensemble MCP, isolant, plaque en cuivre et mullite, est non seulement la plus influente sur la température de la carte d'acquisition mais également la plus critique à modéliser. Un simple sous-système MBSE n'étant pas suffisant, la stratégie MHMN développée dans cette thèse propose de le remplacer par un métamodèle basé sur une hypersurface NURBS entraînée par des données haute-fidélité. Ainsi, l'efficacité et la souplesse du modèle MBSE initial sont conservées tout en lui apportant un degré de précision supplémentaire. Cette découpe MBSE - NURBS est illustrée dans la Fig. 5.12a avec les connecteurs renseignés.

Les données d'entraînement \mathbf{Q}_t et de contrôle \mathbf{Q}_c sont générées à partir d'un modèle axisymétrique aux EF de la zone inférieure illustrée dans la Figure 5.12a. Pour rappel, le nombre de Biot de la plaque en cuivre est inférieur à l'unité. En conséquence, la source de chaleur est appliquée sur le cuivre sous forme de source volumique, avec une puissance de $P = 100$ W appliquée sur une période de 20 s. Du point de vue thermique, les conditions aux limites du système sont modélisées comme adiabatiques, car le processus se déroule sur une courte période de temps et que la mullite présente une faible conductivité thermique. En ce qui concerne les conditions aux limites mécaniques, seules les translations le long de l'axe x sont autorisées sur le bord supérieur, tandis que seules les translations le long de l'axe y sont autorisées sur l'axe de symétrie. Les autres bords du modèle sont considérés comme libres.

Cette modélisation a été réalisée au moyen du logiciel commercial Abaqus. Plus précisément, une analyse couplée thermomécanique non-linéaire a été menée sur le maillage axisymétrique illustré dans la Figure 5.12b. Ce maillage est composé de 64×68 éléments couplés thermomécaniques CAX6MT, chacun possédant six nœuds. Chaque nœud est caractérisé par deux degrés de liberté mécaniques (déplacements u_x et u_y) et un degré de liberté thermique (température T). Ce maillage est issu d'une étude de convergence re-

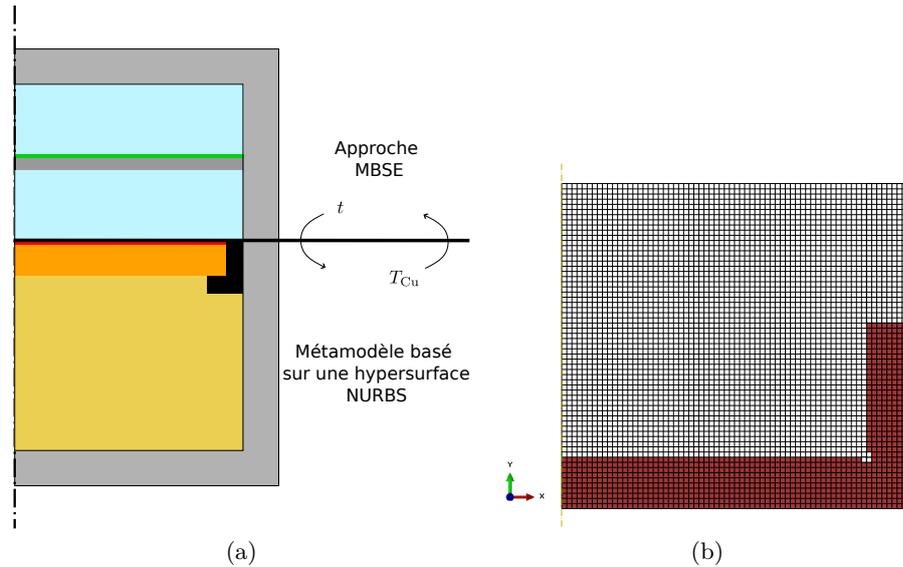


FIGURE 5.12 – Mise en place d’une modélisation via un métamodèle haute-fidélité basé sur une hypersurface NURBS sur la zone du Matériau à Changement de Phase, de la rondelle d’isolant et de la portion de mullite associée, en (a). En (b), visualisation du maillage de l’ensemble du sous-modèle mullite, Matériau à Changement de Phase, isolant, cuivre.

portée en annexe E.4.

L’analyse temporelle s’étend sur la durée totale de la mission ($t_{tot} = 100$ s), au cours de laquelle les états transitoires des températures et des déplacements des nœuds sont observés. Sur le plan thermique, le problème est non-linéaire en raison de la forme en créneau de la source de puissance mais également des températures atteintes, plaçant le MCP dans un état de transition de phase, état qui est fortement non-linéaire. De plus, sur le plan mécanique, le problème est non-linéaire en grandes déformations et grands déplacements, étant donné que la dilatation du MCP est significative (environ 10 %).

La difficulté à présent est de définir les variables d’entrée et les sorties d’intérêts du métamodèle, découlant du modèle aux EF. Elles doivent permettre à la fois de dimensionner le sous-système du métamodèle résultant vis-à-vis du modèle global MBSE, mais également d’assurer la communication avec les variables du modèle MBSE. La zone supérieure du modèle global sur la Fig. 5.12a est modélisée par une approche MBSE purement thermique. Plus précisément, son comportement est régi par la température de la carte de calcul. En outre, cette partie MBSE évolue selon un pas de temps fixé lors du paramétrage du solveur. Pour assurer la communication entre les deux zones, le métamodèle doit fournir au modèle MBSE la température du cuivre tandis que le modèle MBSE doit fournir le pas de temps auquel sont effectués les calculs. Ces variables de communications sont renseignées dans le Tab. 5.4 tout comme les variables de conception. Dans la mesure où l’objectif est de calibrer les différents sous-systèmes du modèle hybride haute-fidélité pour assurer un comportement thermomécanique acceptable, le choix est fait de considérer

comme variable de conception du métamodèle les conductivités du MCP et de la mullite notées respectivement λ_{MCP} et λ_{Mu} , ainsi que leurs épaisseurs e_{MCP} et e_{Mu} .

Quant aux sorties d'intérêt, mise à part la température du cuivre requise en entrée du modèle MBSE, il est nécessaire de considérer la contrainte maximale subie par l'enveloppe en mullite pour contrôler sa tenue mécanique tout au long de la mission. Cette contrainte maximale correspond à la contrainte maximale de Von Mises mesurée sur la zone en rouge dans le maillage de la Fig. 5.12b. Cette zone permet d'écarter de l'étude le bruit numérique engendré par les conditions limites du maillage autour de l'isolant et l'angle droit entre le MCP et la mullite. Les autres sorties d'intérêt permettent d'observer le comportement du MCP au sens large : la température moyenne du volume de MCP notée T_{MCP} et sa variation totale d'énergie interne ΔE_{int} . Pour finir le rapport $\text{Rdm} = \frac{\Delta E_{\text{int}}}{E_{\text{tot}}}$ est calculé pour évaluer l'efficacité du MCP à absorber l'énergie dissipée totale $E_{\text{tot}} = P_{\text{src}} t_{\text{tot}}$.

TABLE 5.4 – Variables d'entrée (*input*) et de sortie (*outputs*) du métamodèle. La variable d'entrée de temps t , est directement reliée à la valeur du flux dissipé par la source.

	Variable	Sortie
Communication	t [s]	T_{Cu} [K]
Conception	λ_{MCP} [$\text{W m}^{-1} \text{K}^{-1}$]	$\sigma_{\text{max,Mu}}$ [Pa]
	λ_{Mu} [$\text{W m}^{-1} \text{K}^{-1}$]	ΔE_{int} [J]
	e_{MCP} [m]	T_{MCP} [K]
	e_{Mu} [m]	Rdm [-]

Les différentes propriétés thermomécaniques du système embarqué sont renseignées dans le Tab. 5.5. Ces données sont issues de la littérature et plus particulièrement de [58, 115] en ce qui concerne la mullite, avec une porosité de 32 %.

TABLE 5.5 – Grandeurs thermiques et mécaniques requises pour la simulation aux EF. S est employé pour *état solide* et L pour *état liquide*.

Grandeur	Symbole	Cuivre	MCP	Mullite 32 %
Masse volumique [kg m^{-3}]	ρ	8954	S : 880, L : 770	3000
Capacité thermique [$\text{J kg}^{-1} \text{K}^{-1}$]	c_p	383	2000	800
Conductivité thermique [$\text{W m}^{-1} \text{K}^{-1}$]	λ	386	2.5	0.6
Chaleur latente [J kg^{-3}]	ΔH	1083	174000	1500
Module d'Young [Pa]	Y	124×10^9	6×10^7	27×10^9
Dilatation [m K^{-1}]	α	17×10^{-6}	1.25×10^{-4}	7×10^{-6}
Contrainte à la rupture [Pa]	σ_{max}	-	-	45×10^6

Le comportement fortement non-linéaire de l'ensemble cuivre, MCP, isolant, mullite est représenté dans la Fig. 5.13 par l'évolution non-homogène du champ de déplacement du MCP et de la contrainte de Von Mises. Dans la partie de gauche, on peut voir l'évolution de l'amplitude du champ de déplacement du MCP, en considérant ce dernier non contraint par l'enveloppe en mullite. Dans la partie de droite, l'ensemble du modèle aux EF est considéré, mais la zone de mesure de la contrainte de Von Mises dans la mullite

est isolée pour une meilleure visualisation. La contrainte maximale dans la mullite est non-linéaire et non-homogène, ce qui confirme l'intérêt d'utiliser un modèle initial aux EF pour la calculer avec une précision acceptable. Vu le nombre de paramètres considérés, les non-linéarités matériaux et l'évolution non-uniforme des observables, il est pertinent de créer un métamodèle capable d'évaluer rapidement un nombre plus important de solutions constructives afin de répondre au mieux au cahier des charges.

5.4.2 Génération du métamodèle comme solution d'un problème d'optimisation

Le métamodèle considéré dans la modélisation du SETC se base sur une hypersurface NURBS évoluant dans un espace MIMO de $N = 5$ dimensions d'entrée et $M = 5$ dimensions de sortie. La génération de ce métamodèle se pose comme la résolution de trois problèmes d'optimisation sous-contraintes successifs. Ce processus est détaillé au chapitre 3 mais consiste, en substance, à :

1. générer les coordonnées des CPs optimales à partir de la base d'entraînement \mathbf{Q}_t . Le métamodèle résultant correspond à une hypersurface B-spline optimisée ;
2. optimiser les valeurs non-triviales des vecteurs nodaux. Le métamodèle résultant correspond à une hypersurface B-spline optimisée par rapport aux coordonnées de ses CPs et à ses vecteurs nodaux ;
3. optimiser les poids associés à chaque CPs. Le métamodèle résultant correspond à une hypersurface NURBS optimisée par rapport à ses variables de conception continues.

A chacune de ces étapes, la précision du métamodèle est évaluée à partir d'une base \mathbf{Q}_c , différente de la base d'entraînement \mathbf{Q}_t . Le métamodèle optimal peut être issu aussi bien de la première, seconde ou dernière étape du processus. Les indicateurs employés sont :

— le coefficient

$$R_\alpha^2 = 1 - \sum_{i=1}^{n_{\text{TP}}} \frac{(h_\alpha(\zeta_{i_1}, \dots, \zeta_{i_N}) - q_{i_\alpha})^2}{(q_{i_\alpha} - \bar{q}_\alpha)^2},$$

pour chaque direction de sortie α ;

— l'erreur relative

$$\bar{\varepsilon}_\alpha = \frac{1}{n_{\text{TP}}} \sum_{i=1}^{n_{\text{TP}}} \frac{(h_\alpha(\zeta_{i_1}, \dots, \zeta_{i_N}) - q_{i_\alpha})}{q_{i_\alpha}},$$

pour chaque direction de sortie α .

Compte tenu des temps de calculs relativement importants d'Abaqus (environ 7 min par simulation, pour calculer les solutions toutes les secondes durant 100 s), la quantité de données disponible est relativement limitée. Ainsi, le nombre de TPs considérés dans la base de données d'entraînement est de $n_{\text{TP},t} = 122451$ et le nombre de CPs de $n_{\text{CP},t} = 42768$. Quant à la base de données de contrôle, elle comporte $n_{\text{TP},c} = 12544$ TPs. La base de données globale dont sont issues les bases de données d'entraînement et de contrôle

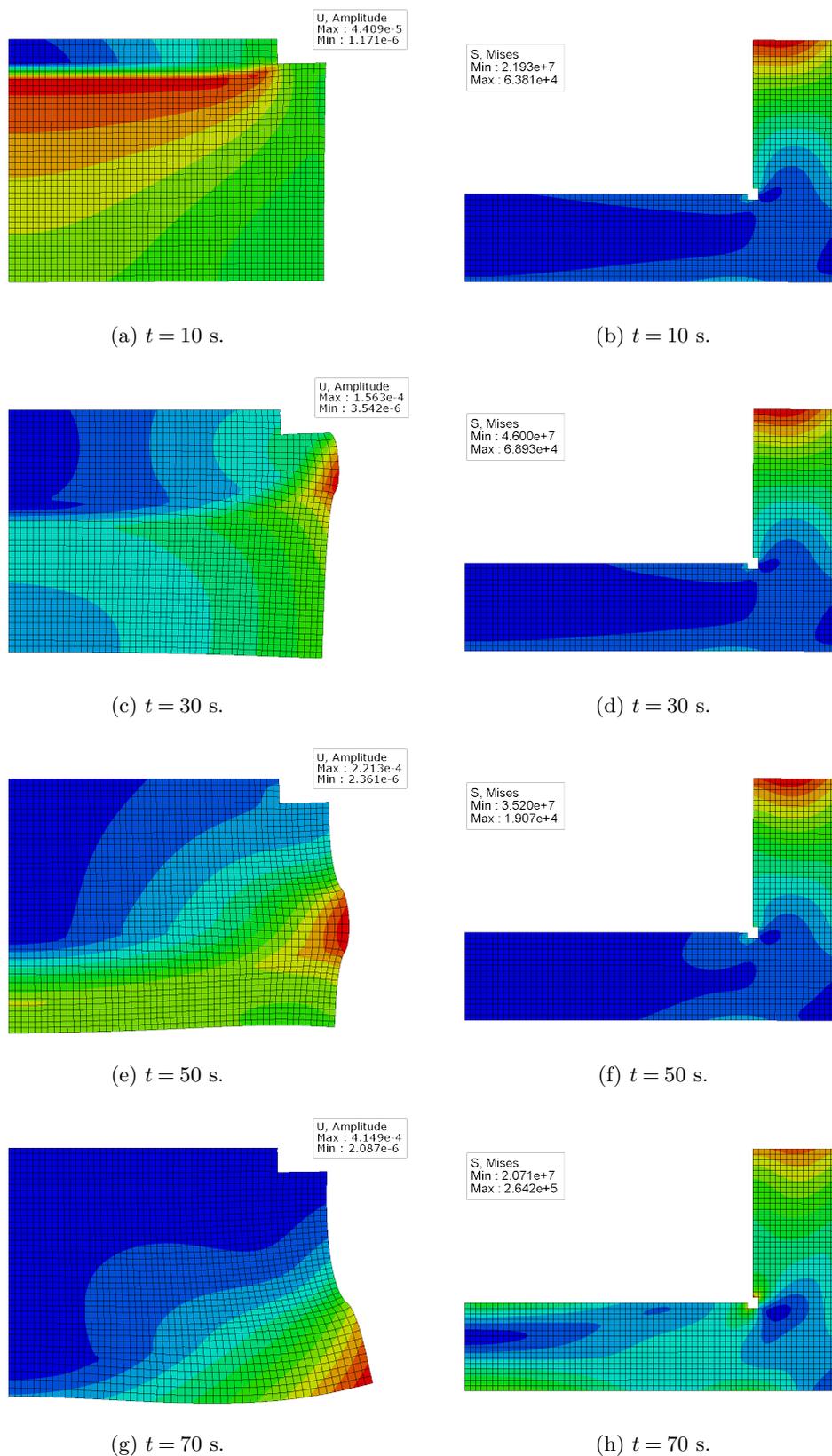


FIGURE 5.13 – Évolution temporelle du champ de déplacement (amplitude en mm) pour la région caractérisée par le MCP dans (a), (c), (e) et (g) et de la contrainte de Von Mises (en Pa) dans la mullite en (b), (d), (f) et (h).

est construite à partir de 10000 simulations via 10 licences Abaqus en parallèles pour un temps de simulation d'environ une semaine. Les valeurs des différents paramètres du métamodèle sont renseignées dans le Tab. 5.6. On y retrouve également les plages de variation des différentes variables de conception, réparties selon une loi uniforme notée $\mathcal{U}[\cdot, \cdot]$, ainsi que les intervalles de variation des variables d'optimisation β_i et ω_{i_1, \dots, i_N} (se référer au chapitre 3 pour plus de détails).

TABLE 5.6 – Caractéristiques de l'hypersurface NURBS et des bases de données associées.

Variable	Intervalle	s_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
t	$\mathcal{U}([0.0, 100.0])$	2	50	32				
λ_{MCP}	$\mathcal{U}([2.0, 4.0])$	2	6	5				
λ_{Mu}	$\mathcal{U}([0.45, 0.75])$	2	6	5	122451	20625]0, 1[[1, 10]
e_{MCP}	$\mathcal{U}([5.0, 15.0]) \times 10^{-3}$	2	6	5				
e_{Mu}	$\mathcal{U}([2.0, 6.0]) \times 10^{-3}$	2	6	5				
\mathbf{Q}_c	–	(3, 3, 3, 3, 48)			12544	–	–	–

5.4.3 Précision du métamodèle

Les valeurs des différents indicateurs d'efficacité, f_{obj} , et de précision, R_α^2 et $\bar{\varepsilon}_\alpha$, de la génération du métamodèle sont rassemblées dans le Tab. 5.7. Plus précisément, ces valeurs correspondent aux moyennes des indicateurs sur les sorties α du problème. On peut constater que les différentes optimisations n'ont que peu d'influence. A la fin du processus, elles sont de : $f_{obj} = 0.98$, $R_{glob}^2 = 0.99$ et $\bar{\varepsilon}_{glob} = 1.13$ %. La Fig. 5.14b illustre cela en traçant les valeurs calculées par le métamodèle en fonction des valeurs de la base de données d'entraînement en considérant toutes les sorties. La majorité des données sont confondues avec la courbe de référence $x = y$. Le graphique de *fitting* dans la Fig. 5.14a considère seulement la sortie la plus critique, à savoir l'évaluation de la contrainte maximale de Von Mises dans la mullite. En conclusion, les indicateurs $R_{\sigma_{max, Mu}}^2 = 0.90$ et $\bar{\varepsilon}_{\sigma_{max, Mu}} = 5.32$ % restent acceptables et le métamodèle peut être employé dans la stratégie MHMN.

 TABLE 5.7 – Évolution des coefficients f_{obj} , R_{glob}^2 et $\bar{\varepsilon}_{glob}$, en fonction de l'étape du processus de génération du métamodèle.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$	$\xi_{1,opt}$	$\xi_{2,opt}$	$\xi_{3,opt}$
1	0.98	0.98	0.9956	0.9975	0.9975	1.12	1.13	1.13

Un des avantages de ce métamodèle est sa rapidité d'exécution, permettant de le coupler au modèle global MBSE sans difficulté. Le temps de simulation du modèle Abaqus pour obtenir les courbes de comportement sur $\Delta t = 100$ s est de 420 s, là où le métamodèle a besoin de seulement 3 s. En d'autres termes, le métamodèle est capable de calculer les

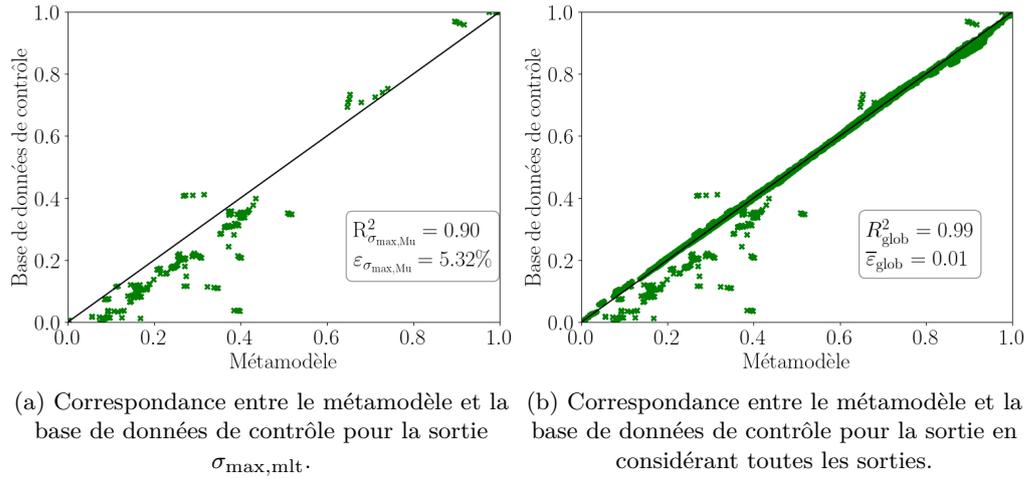


FIGURE 5.14 – Correspondance entre le métamodèle et la base de données de contrôle.

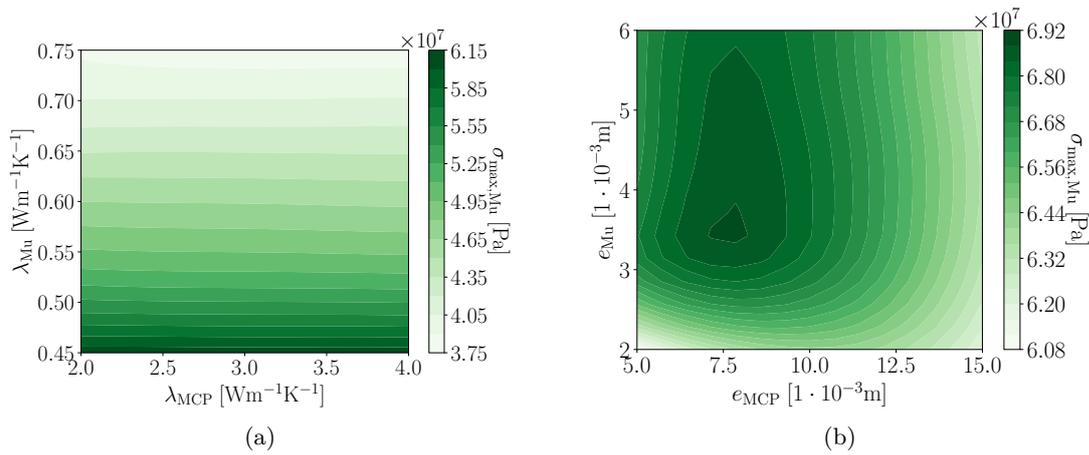


FIGURE 5.15 – Contrainte maximale subie par la mullite au cours de la simulation en fonction des paramètres du métamodèle : en (a) e_{MCP} et e_{Mu} sont fixes et en (a) λ_{MCP} et λ_{Mu} .

sorties en 0.03 s par pas de temps, lui permettant de communiquer en temps réel dans la stratégie complète MHMN.

En outre, une fois le métamodèle généré, il peut être manipulé à la guise de l'utilisateur en fonction de son besoin. Par exemple, la Fig. 5.15 trace les courbes de niveau de la contrainte maximale subie par la mullite sur l'ensemble de la simulation en fonction des autres paramètres : dans la Fig. 5.15a, en considérant les différents coefficients de conductivité et dans la Fig. 5.15b les épaisseurs du MCP et de la mullite. Dans les deux situations, les paramètres fixes sont définis à leur valeur de référence renseignées dans le Tab. 5.5. On peut constater que $\sigma_{\max,Mu}$ est plus importante pour des épaisseurs fines.

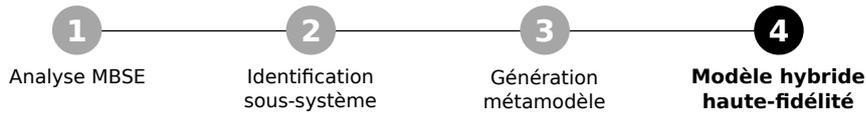


FIGURE 5.16 – Quatrième étape de la stratégie Modélisation Hybride MBSE - NURBS : mise en place complète.

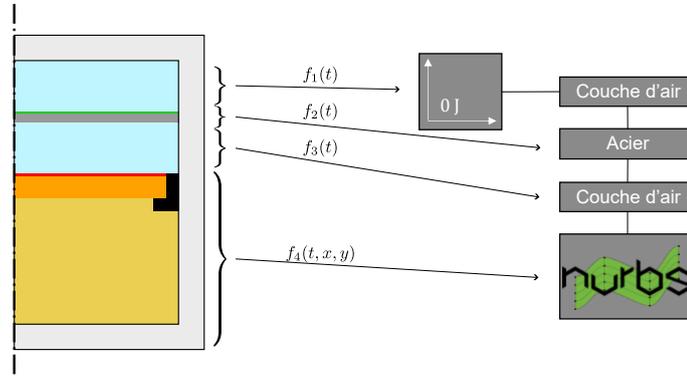


FIGURE 5.17 – Schéma de l'inclusion du métamodèle basé sur les entités NURBS dans le modèle global *Model-Based System Engineering* du système embarqué incluant un Matériau à Changement de Phase.

5.5 Stratégie hybride haute-fidélité

Comme illustré dans la Fig. 5.16, la mise en place complète de la stratégie de modélisation hybride haute-fidélité se déroule en quatre étapes. La quatrième, réalisée dans cette section, consiste à mettre en place la stratégie complète sur le modèle d'étude.

5.5.1 Construction du modèle hybride haute-fidélité du SETC-3

Cette section propose la mise en place de la stratégie MHMN développée tout au long de ce manuscrit. Elle consiste à intégrer dans le modèle MBSE du SETC-2, le métamodèle basé sur les hypersurfaces NURBS construit à la section 5.4, à la place de la zone critique et influente identifiée à la section 5.3 correspondant au groupement MCP, isolant, cuivre et mullite. La Fig. 5.17 illustre la mise en place de cette modélisation sur le problème traité.

5.5.2 Initialisation du modèle hybride haute-fidélité

Pour rappel, l'objectif de ce chapitre est de dimensionner un système embarqué de mesure en respectant ses conditions de fonctionnement sur une durée totale de $t_{\text{tot}} = 100$ s. La première exigence est de restreindre la montée en température de la carte électronique d'acquisition en dessous de $T_{\text{max,adm}} = 373$ K. La seconde exigence est que la contrainte maximale dans la mullite ne dépasse pas le seuil de rupture, fixé à $\sigma_{\text{max,Mu}} = 45 \times 10^6$ Pa [58, 115].

Pour dimensionner au mieux ce dispositif, le processus classique suivant est mis en place :

1. une première simulation est faite pour avoir une solution de référence, en considérant les valeurs de référence des différentes variables de conception ;
2. une ASG de type OAT [88] est appliquée sur le modèle pour déterminer les tendances de comportement des sorties d'intérêt en fonction des valeurs extrêmes des différentes variables de conception. Les différents jeux de paramètres en question sont rassemblés dans le Tab. 5.8 ;
3. un jeu de valeur des différentes variables de conception est choisi comme solution du problème.

TABLE 5.8 – Jeux de paramètres des différentes simulations considérées dans le dimensionnement du système embarqué.

Variable de conception	λ_{MCP}	λ_{Mu}	e_{MCP}	e_{Mu}
Simulation de référence	2,5	0,60	0,010	0,004
Simulation 1	2.0	0.60	0.010	0.004
Simulation 2	4.0	0.60	0.010	0.004
Simulation 3	2.5	0.45	0.010	0.004
Simulation 4	2.5	0.75	0.010	0.004
Simulation 5	2.5	0.60	0.005	0.004
Simulation 6	2.5	0.60	0.015	0.004
Simulation 7	2.5	0.60	0.010	0.002
Simulation 8	2.5	0.60	0.010	0.006

5.5.3 Résultats du modèle hybride haute-fidélité SETC-3

Les résultats des différentes simulations du processus de dimensionnement sont fournis dans le Tab. 5.9. On peut remarquer que dans toutes les situations la première exigence sur la température de fonctionnement de la carte d'acquisition est respectée. Cependant, ce n'est pas le cas pour la seconde exigence qui porte sur la contrainte mécanique subie par la mullite. Dans la majeure partie des cas, elle est supérieure au seuil acceptable. Néanmoins, on peut remarquer qu'entre les résultats 1 et 2, sa valeur varie de 9 % et entre les résultats 7 et 8 sa valeur varie de presque 60 %. Ces deux jeux de résultats utilisent respectivement les paramètres fixés à leur valeur de référence hormis la conductivité du MCP λ_{MCP} et l'épaisseur de la mullite e_{Mu} .

Ces observations coïncident avec les résultats de l'ASG et permettent de converger vers une solution. Le jeu de valeurs des variables de conception est renseigné au Tab. 5.10. On peut alors constater que la première exigence est toujours respectée : $T_{\max,Acier} = 317.84 < 373$ K. Quant à la seconde, même en considérant l'erreur relative de 5 % commise par

TABLE 5.9 – Solution des différentes simulations considérées dans le dimensionnement du système embarqué pour la contrainte maximale subie par la mullite et la température maximale atteinte par le support en acier.

Sortie d'intérêt	$T_{\max, \text{Acier}}$	$\sigma_{\max, \text{Mu}}$
Résultat de référence	323.81	$5,14 \times 10^7$
Résultat 1	329.69	5.41×10^7
Résultat 2	323.80	4.96×10^7
Résultat 3	323.86	5.15×10^7
Résultat 4	323.60	5.13×10^7
Résultat 5	329.01	4.83×10^7
Résultat 6	324.12	4.90×10^7
Résultat 7	324.49	7.96×10^7
Résultat 8	315.71	4.71×10^7

TABLE 5.10 – Solution choisie pour respecter les exigences de fonctionnement du Système Embarqué à Température Contrôlée.

	λ_{MCP}	λ_{Mu}	e_{MCP}	e_{Mu}	$T_{\max, \text{Acier}}$	$\sigma_{\max, \text{Mu}}$
Solution	3.5	0.70	0.014	0.006	317.84	4.14×10^7

le métamodèle sur le modèle initial, elle est également respectée : $\sigma_{\max, \text{Mu}} = 4.14 \times 10^7 \pm 2.07 \times 10^6 < 4.5 \times 10^7$ Pa.

Un autre intérêt de cette stratégie MHMN est l'accès à toutes les variables entrant en compte dans les lois de comportement des différents sous-systèmes. Il est possible d'accéder à d'autres solutions que celles nécessaires au bon fonctionnement du système embarqué. Ainsi, en considérant les valeurs de solution du Tab. 5.10, la variation d'énergie interne du MCP est de $\Delta E_{\text{int}} = 668$ J. Le rendement du MCP à absorber l'énergie dissipée par la carte de calcul est alors de $\text{Rdm} = \frac{\Delta E_{\text{int}}}{P_{\text{calc}} t_{\text{tot}}} = 33$ %. Les Figs. 5.18a 5.18b et 5.18c, tracent respectivement les évolutions des températures de l'acier, du cuivre et du MCP. On peut constater que la valeur maximale atteinte par la carte de calcul est de $T_{\max, \text{Cu}} = 519$ K et celle du MCP de $T_{\max, \text{MCP}} = 390$ K. Ces valeurs sont en dessous des plages de travail des deux matériaux, à savoir de la température de fusion du $T_{\text{fus}, \text{Cu}} = 1356$ K et du point d'auto-combustion du MCP de $T_{\text{comb}, \text{MCP}} = 573$ K.

La stratégie MHMN a permis de modéliser efficacement le dispositif SETC avec une description adaptée au besoin de précision. Si ce système doit à l'avenir faire l'objet d'une complexification par ajout de composants internes ou externes, il sera possible de le faire via l'aspect MBSE de la méthode. Le problème pourra alors évoluer en fonction de la demande, sans pour autant que les coûts de calculs s'accroissent. D'autres métamodèles pourront aussi être intégrés. Cependant, bien que le temps de simulation MHMN soit de l'ordre de la seconde, la génération du métamodèle basé sur les entités NURBS peut être plus fastidieuse en fonction de la situation. Dans le cas d'étude présenté, la génération du métamodèle lui-même a nécessité aux environs de 2.5 h mais la construction des bases de

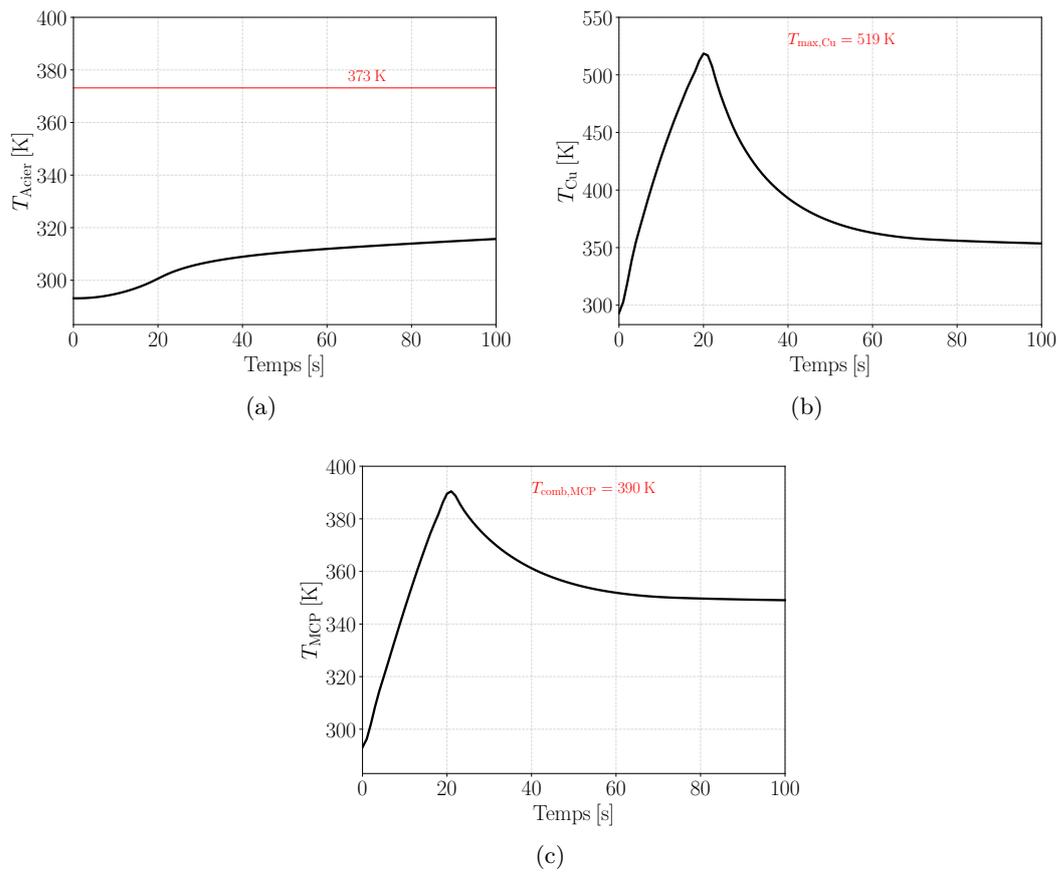


FIGURE 5.18 – Évolution au cours du temps des températures de la plaque en acier (a), en cuivre (b) et du Matériau à Changement de Phase (d).

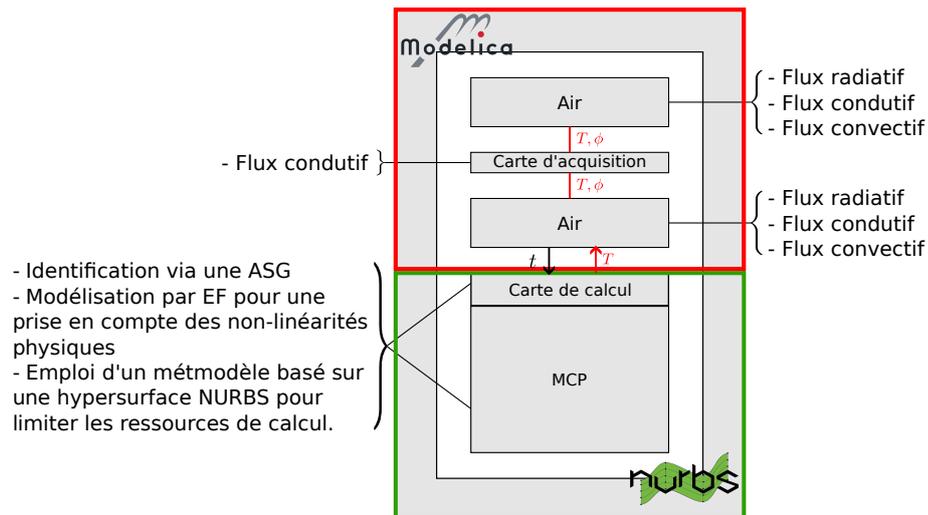


FIGURE 5.19 – Schéma fonctionnel final du Système Embarqué à Température Contrôlée avec identification des zones modélisées par approche *Model-Based System Engineering* et par le métamodèle basé sur une hypersurface NURBS.

données d'entraînement et de contrôle, aux environs d'une semaine (10000 simulations). Le choix des variables d'entrée du métamodèle n'est donc pas anodin et demande une attention particulière. Notons que ce choix est nécessaire qu'une fois dans un cycle de conception *V-and-V* et doit être fait par des spécialistes. Une fois fait, le modèle peut être transféré vers le bureau d'étude pour des études paramétriques, par exemple. Il pourra même être utilisé pour des questions d'optimisation de performance.

5.6 Conclusion

La stratégie MHMN développée dans cette thèse a permis de dimensionner un modèle de Système Embarqué à Température Contrôlée. L'objectif était de déterminer les caractéristiques des différents matériaux constitutifs pour assurer son bon fonctionnement. Dans un premier temps, une étude MBSE a été menée pour identifier les tendances d'évolution des différents points d'intérêt du modèle. L'emploi d'un métamodèle basé sur une hypersurface NURBS a ensuite permis d'apporter des informations supplémentaires au modèle global MBSE pour répondre aux exigences de dimensionnement inhérentes au système embarqué étudié. La Fig. 5.19 fait une synthèse de l'application de cette stratégie sur le modèle SETC.

Il ne s'agit que d'un exemple d'application, pour montrer que ce type de stratégie pourra être appliqué à des cas plus complexes par la suite, en considérant l'intégration de plusieurs métamodèles. Les principales perspectives d'évolution de cette stratégie, en plus d'améliorer la robustesse du métamodèle, sont de développer une méthode d'identification objective et systématique des sous-systèmes critiques et/ou influents, d'identifier automatiquement les variables de communication entre le métamodèle et le modèle MBSE ainsi

que de considérer la propagation des différentes sources d'incertitudes. En outre, dans la mesure où les formulations des sous-systèmes restent analytiques (avantage du métamodèle basé sur une hypersurface NURBS), le calcul du gradient de l'ensemble du modèle hybride pourrait permettre d'optimiser ses variables de conception de manière objective.

Conclusion

Ce manuscrit a exposé la mise en place de la stratégie MHMN développée au cours de ce travail de thèse qui permet une modélisation multiphysique optimisée, avec le juste niveau de précision apporté à chaque sous-système du modèle global. Après avoir effectué un état de l'art qui comprend notamment une présentation de l'approche MBSE, le deuxième chapitre a proposé un tour d'horizon des méthodes d'ASG utilisées dans cette étude. Ces méthodes ont été appliquées à des cas d'intérêt modélisés à la fois via l'approche MBSE et par le biais d'analyses aux EF. L'objectif de l'ASG est d'apporter un soutien à l'évaluation critique menée par le bureau d'étude pour identifier les sous-systèmes nécessitant une description approfondie qu'une approche MBSE classique ne permet pas. L'objectif est ensuite d'approfondir leur description par l'emploi d'un métamodèle basé sur les hypersurfaces NURBS.

Dans le troisième chapitre, nous abordons la création de ce métamodèle, conçu comme une solution à un problème d'optimisation sous contraintes. L'utilisation d'hypersurfaces NURBS présente un intérêt particulier en raison de leur grande adaptabilité, qui permet de mieux représenter les non-linéarités des problèmes traités. Pour y parvenir, les variables de conception continues des hypersurfaces sont déterminées par la résolution successive de CNLPP, spécifiques à chaque variable. Le quatrième chapitre s'attache à évaluer l'efficacité du métamodèle dans divers scénarios, allant de problèmes peu non-linéaires à fortement non-linéaires, et de cas théoriques analytiques à des problèmes plus pratiques modélisés par la méthode des EF.

Finalement, le dernier chapitre applique la stratégie MHMN à un cas d'intérêt du CEA, impliquant une grande variété de comportements thermomécaniques non-linéaires pouvant être rencontrés dans les problèmes du CEA. Cette stratégie est intéressante dans la mesure où, en un seul modèle, il est possible de représenter l'ensemble des comportements du problème traité, quelle que soit la physique en jeu. En outre, les zones nécessitant une précision accrue peuvent être modélisées par un métamodèle haute-fidélité, aussi peu gourmand en ressources de calculs que le modèle analytique MBSE initial. La modularité permise par la stratégie MHMN peut permettre ensuite de compléter efficacement le modèle global, si besoin.

Le verrou scientifique identifié en amont de ce travail de thèse était de parvenir à développer la stratégie de métamodélisation basée sur les hypersurfaces NURBS comme solution d'un problème d'optimisation sous contraintes. En se basant sur des travaux

antérieurs, un processus d'optimisation en trois étapes a été développé et mis en place. Chacune des étapes constitue un CNLPP associé à une variable de conception continue du métamodèle : (i) les coordonnées des CPs, évaluées par une résolution analytique du CNLPP, (ii) les valeurs non triviales des vecteurs nodaux, évaluées par résolution numérique d'un CNLPP via un algorithme au gradient GC-MMA et (iii) les poids associés aux CPs, issus également de la résolution numérique d'un CNLPP. Dans ce cadre, cette thèse apporte des contributions significatives en formulant analytiquement le gradient de la fonction de coût pour les étapes (ii) et (iii) de la stratégie d'optimisation, en considérant les contraintes associées. Elle introduit également un terme de lissage qui dépend des dérivées partielles d'ordre 2 de l'hypersurface : l'objectif est de minimiser le phénomène d'*overfitting* et de restreindre la dépendance de la solution au bruit numérique.

Le second verrou identifié était technique : parvenir à inclure le métamodèle dans un modèle global MBSE, après avoir identifié le sous-système nécessitant un effort de calcul renforcé. Cette identification se base sur l'avis critique d'un bureau d'études, appuyé par la mise en place d'une ASG. Une fois le sous-système identifié et le métamodèle associé généré, ce dernier est inclus dans le modèle global via une interface Python - C - Modelica mis en place par [167]. Le lien Python - C n'est pas trivial mais l'appel du C par Modelica fait partie des avantages intrinsèques de ce langage dont la sémantique se base sur celle du C.

Les avantages apportés dans la génération du métamodèle basé sur les entités NURBS par ces optimisations successives sont nombreux. La détermination des coordonnées des CPs par résolution du CNLPP associé permet de respecter la densité de répartition des données d'entraînement. L'ajout d'un terme de lissage permet de réduire l'effet des éventuelles non-linéarités parasites de la base de données sur le métamodèle ainsi que le phénomène d'*overfitting*. L'optimisation des valeurs non triviales des vecteurs nodaux permet de pleinement tirer avantage de la propriété de support local des hypersurfaces NURBS : cela apporte de la maniabilité au métamodèle dans les zones de non-linéarité. Pour finir, l'optimisation des poids permet d'intervenir sur l'hypersurface sans modifier les autres grandeurs déjà optimisées. Ainsi, une fois les CPs fixés, leur poids est modifié pour leur donner plus ou moins d'importance en fonction de la présence de non-linéarités. Les résolutions approchées numériques des deux derniers CNLPP sont mises en place via l'algorithme au gradient employant la méthode GC-MMA. Cette méthode est robuste et permet notamment de considérer un grand nombre de variables d'optimisation mais aussi de prendre en compte des initialisations en dehors du domaine d'étude. En outre, une fois généré, le métamodèle permet un gain de temps considérable dans la simulation du problème étudié. Quel que soit son nombre de dimensions, le métamodèle fournira une réponse quasi-instantanée, ce qui est primordial pour avoir une communication en temps réel avec le modèle global MBSE.

La stratégie MHMN présentée dans ce travail, est une innovation apportée au domaine

de la modélisation et de la conception de systèmes complexes. Cette approche combine les avantages de l'approche MBSE avec l'utilisation de métamodèles basés sur les hypersurfaces NURBS. L'objectif ultime de la stratégie MHMN est de compléter et d'approfondir les études de pré-dimensionnement menées en amont des projets, qu'ils soient réalisés par un bureau d'études, plus technique, ou des architectes produit, plus théoriques.

La stratégie MHMN bien qu'innovante et polyvalente présente une limitation dans l'objectivité de l'identification des sous-systèmes critiques, correspondant au second verrou de ce travail. En effet, l'ASG permet d'identifier les paramètres influents sur la sortie considérée mais un avis critique d'utilisateur doit pouvoir en déduire quel sous-système est le plus influent. Cette tâche est loin d'être triviale dans la mesure où le choix des variables employées dans l'analyse est crucial : si les variables ont un rôle dans plusieurs sous-systèmes, il sera compliqué de déterminer lequel nécessite une description approfondie. De même, si parmi les variables retenues pour l'analyse il n'y a pas de variables influentes, le résultat en sera biaisé.

Cette constatation constitue une perspective intéressante à explorer pour améliorer la stratégie MHMN. La solution serait d'employer une méthode pouvant à la fois prendre en compte toutes les variables de conception du modèle MBSE initial, tout en ayant une information sur leur rôle dans les différents sous-systèmes du modèle global. En parallèle, le développement d'un indicateur objectif sur le niveau de précision requis par sous-système pourrait aider à calibrer le métamodèle employé ou à conserver la description par EAD initiale.

Pour ce qui est du métamodèle basé sur les hypersurfaces NURBS, il pourrait être intéressant de prendre en compte la ré-évaluation des coordonnées des CPs lors de l'optimisation des poids qui leur sont associés. Il serait alors possible d'intégrer à la troisième étape du processus de génération du métamodèle le terme de lissage. Les éventuelles non-linéarités parasites de la base de données d'entraînement seraient alors réduites plus efficacement et les risques d'*overfitting* complètement écartés. Une autre difficulté à lever, commune pour la plupart des méthodes de métamodélisation, est de parvenir à recouvrir l'espace des possibles en grande dimension avec le moins de ressources possibles. Plusieurs études sur des méthodes de PE sont menées mais la solution la plus viable à ce jour est l'emploi d'une méthode adaptative : des points de données sont ajoutés dans les zones de plus grande erreur. Pour que cela soit efficace, un processus gaussien pourrait être ajouté à la formulation analytique du problème d'optimisation dont est issue le métamodèle développé dans ce travail. Ce métamodèle fonctionnerait de la même manière qu'un processus de krigeage mais en employant les entités NURBS comme fonction de base. Une autre solution intéressante serait d'employer un PE pour construire efficacement les bases de données d'entraînement et de contrôle, pour ne pas faire appel au modèle initial pendant la génération du métamodèle, i.e., pour que la génération du métamodèle soit autonome. Les PE étudiés actuellement au laboratoire I2M sont les méthodes de sparse-grid [195],

de *LATIN HyperCube* (LHC) [170] et de Sobol [134].

Une autre perspective pour la stratégie complète MHMN est de déterminer objectivement les points de communication entre le modèle global MBSE et le métamodèle basé sur une hypersurface NURBS. Ces variables de communication sont primordiales dans la mesure où elles permettent de lier l'ensemble du modèle hybride lors de la simulation. Dans ce travail, leur choix s'est argumenté autour de plusieurs critères (variables de potentiel ou de flux, temporalité, physique en jeu) mais reste tout de même subjectif. Une piste à explorer est l'emploi d'une méthode d'homogénéisation pour réduire à une variable un gradient fournit par une analyse EF.

Ce travail de thèse a permis d'étudier en détail la génération d'un métamodèle basé sur des hypersurfaces NURBS, en examinant ses avantages et inconvénients. Il a été réalisé dans un contexte de pré-dimensionnement, visant à maintenir une précision élevée sans compromettre l'efficacité des simulations. Cette approche a été partiellement appliquée aux projets du bureau d'études du CEA, en se concentrant sur des systèmes thermomécaniques complexes bien que des problèmes hydromécaniques soient également à l'étude.

A

Annexe - Introduction générale

A.1 Valorisation de la Communication entre MBSE et CATIA dans un Contexte Hydro-mécanique

Le bureau d'études du CEA conçoit des systèmes multiphysiques par le biais de logiciels indépendants tels que CATIA ou Abaqus, ou via des codes développés en interne. Le stage encadré durant cette thèse a permis d'apporter un regard nouveau sur le pré-dimensionnement des produits développés au CEA. Cette annexe résume les principaux aspects du stage encadré durant cette thèse, visant à étudier les possibilités et les limites de la communication entre l'approche MBSE basée sur le langage Modelica, et le logiciel de conception 3-D CATIA, dans le contexte d'un problème hydromécanique. En outre, il s'agissait d'étudier la viabilité et l'utilité du logiciel MBSE Dymola de la plateforme 3D Experience de Dassault System pour le bureau d'étude du CEA.

Les objectifs de ce stage comprenaient la compréhension des besoins de communication, l'identification des défis spécifiques, la proposition de solutions et la démonstration de leur efficacité.

Le stage a couvert l'ensemble de l'élaboration du modèle de pré-dimensionnement hydromécanique étudié. Une comparaison entre différents logiciels MBSE a permis de retenir le logiciel Dymola pour son inclusion dans la plateforme 3D Experience de Dassault System. Cette particularité a permis ensuite de coupler le modèle MBSE du démonstrateur hydromécanique au modèleur 3-D CATIA. Le premier a apporté un comportement physique au modèle 3-D tandis que le second a participé à l'amélioration de l'ingénierie système qui maîtrise seulement des modèles à faibles degrés de liberté (modèles 0-D ou 1-D).

Ce travail de stage a apporté une nouvelle compétence pour accélérer les procédures de pré-dimensionnement et permettra par la suite de porter un effort de modélisation seulement sur des zones spécifiques. En outre, l'exportation du modèle final au format FMU (fichier exécutable performant) a été étudiée et offre de nouvelles opportunités pour la réalisation d'études statistiques approfondies, renforçant ainsi l'intérêt de l'approche système au sein d'un bureau d'études. Les particularités du format FMU sont exposées

plus en détail au chapitre 1.

B

Annexe - Chapitre 1

B.1 Rappel sur le calcul tensoriel

B.1.1 Algèbre tensorielle en 2 dimensions

Soit $\mathcal{A} = \underline{\underline{A}}$ un tenseur d'ordre 2 indépendant de toute base et $\underline{\underline{T}} = \sum_j^r \underline{\underline{F}}_j \otimes \underline{\underline{G}}_j$ un second tenseur d'ordre 2 correspondant à la somme des produits tensoriels des vecteurs $\underline{\underline{F}}_j$ et $\underline{\underline{G}}_j$. Dans la suite de cette annexe, la notation α correspondra à une grandeur scalaire, $\underline{\underline{F}}$ à un vecteur et $\underline{\underline{G}}$ à une matrice.

B.1.1.1 Produit tensoriel

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (\text{B.1})$$

$$\begin{aligned} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} &= \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \\ &= \underline{\underline{F}}_1 \otimes \underline{\underline{G}}_1 + \underline{\underline{F}}_2 \otimes \underline{\underline{G}}_2 \\ &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{aligned} \quad (\text{B.2})$$

B.1.1.2 Produit scalaire de deux tenseurs

$$\mathcal{A} \odot \underline{\underline{T}} = \sum_{i=1}^2 (\underline{A}_i^x \otimes \underline{A}_i^y) \sum_{j=1}^r (\underline{F}_j \otimes \underline{G}_j) \quad (\text{B.3})$$

B.1.1.3 Produit scalaire, scalaire et tenseur

$$\begin{aligned} \alpha \odot \underline{\underline{T}} &= \alpha \sum_{j=1}^r \underline{F}_j \otimes \underline{G}_j \\ &= \sum_j (\alpha \underline{F}_j) \otimes \underline{G}_j \\ &= \sum (\sqrt{\alpha} \underline{G}_j) \otimes (\sqrt{\alpha} \underline{G}_j) \end{aligned} \quad (\text{B.4})$$

B.1.1.4 Somme tensorielle

$$\begin{aligned} \underline{\underline{T}} \oplus \underline{\underline{T}}' &= \sum_{j=1}^r \underline{F}_j \otimes \underline{G}_j + \sum_{j=1}^{r'} \underline{F}'_j \otimes \underline{G}'_j \\ &= \sum_{j=1}^{r+r'} \underline{F}_j \otimes \underline{G}_j + \underline{F}'_j \otimes \underline{G}'_j \end{aligned} \quad (\text{B.5})$$

B.1.1.5 Produit scalaire

$$\langle \underline{\underline{T}}, \underline{\underline{T}}' \rangle = \sum_{j=1}^r \sum_{j'=1}^{r'} \langle \underline{F}_j, \underline{F}'_{j'} \rangle \langle \underline{G}_j, \underline{G}'_{j'} \rangle \quad (\text{B.6})$$

B.1.2 Opérations

$$a = \begin{pmatrix} a_{i,j} \\ a_{i,j} \\ a_{i,j} \end{pmatrix} \quad (\text{B.7})$$

$$b = \begin{pmatrix} b_{i,j} \\ b_{i,j} \\ b_{i,j} \end{pmatrix}$$

$$a \cdot b = a^T b \quad (\text{B.8})$$

$$a \otimes b = ab^T \quad (\text{B.9})$$

$$(a \otimes b)c = a(b \cdot c) \quad (\text{B.10})$$

B.1.2.1 Exemple d'exploitation

$$\begin{aligned} \frac{\partial \underline{\underline{T}}}{\partial t} - \mathcal{A}\underline{\underline{T}} &= 0 \\ \frac{\underline{\underline{T}}^{n+1} - \underline{\underline{T}}^n}{\Delta t} - \mathcal{A}\underline{\underline{T}}^n &= 0 \\ \underline{\underline{T}}^{n+1} &= \underline{\underline{T}}^n \oplus \Delta t \odot (\mathcal{A} \odot \underline{\underline{T}}^n) \end{aligned} \quad (\text{B.11})$$

B.2 La manipulation symbolique pour le langage Modelica

Le langage Modelica se base sur la sémantique de Kirchhoff pour lier les différents sous-systèmes analytiques d'un modèle MBSE. Pour parvenir à construire un système d'EAD global qui est ensuite résolu par un solveur temporel, Modelica rassemble les différents sous-systèmes EAD par manipulation symbolique, comme illustré dans la Fig. B.1.

La résolution du système EAD global, illustré par la Fig. B.1, se compose de sept étapes liées par six actions. En partant du script initial Modelica, le schéma bloc du modèle est construit par l'utilisateur. Lorsque la résolution du modèle est lancée, le premier travail du logiciel consiste à mettre à plat le modèle, i.e., à faire rassembler tous les morceaux du code répartis dans les différents sous-systèmes. L'aspect "objet" du programme disparaît alors complètement. Une analyse du code résultant est faite dans le but de mettre le système sous forme matricielle avant de l'optimiser et de le résoudre.

Cette étape est la plus longue et trois algorithmes sont nécessaires à sa réalisation : l'algorithme de Pantelides [136], celui des Dummy Derivative présenté par Mattsson dans [120] et l'algorithme de Tarjan [176]. Le rôle du premier est de déterminer l'indice structurel du système EAD à résoudre, à ne pas confondre avec le degré du problème. Le degré d'un système correspond au plus haut degré de dérivation d'une variable. D'un autre côté, un indice structurel est le nombre total de fois qu'il est nécessaire de dériver une ou plusieurs équations du système pour le résoudre. Une fois déterminé, cet indice structurel est réduit à l'aide de l'algorithme de Mattsson jusqu'à une valeur de 0 ou 1, correspondant à celui d'un système d'Équations Différentielles Ordinaires (EDO). Pour faciliter cette étape, l'algorithme de Tarjan est employé pour réduire le nombre d'équations en jeu dans le problème ainsi que le nombre de variables (équation redondante, initialisation d'un paramètre, etc.). Cette manipulation symbolique permet de donner à Modelica son aspect acausal.

Pour finir, le code résultant est traduit en langage C avant d'être compilé et exécuté. Pour de plus amples informations sur le langage Modelica, le lecteur est invité à consulter

le livre [59].

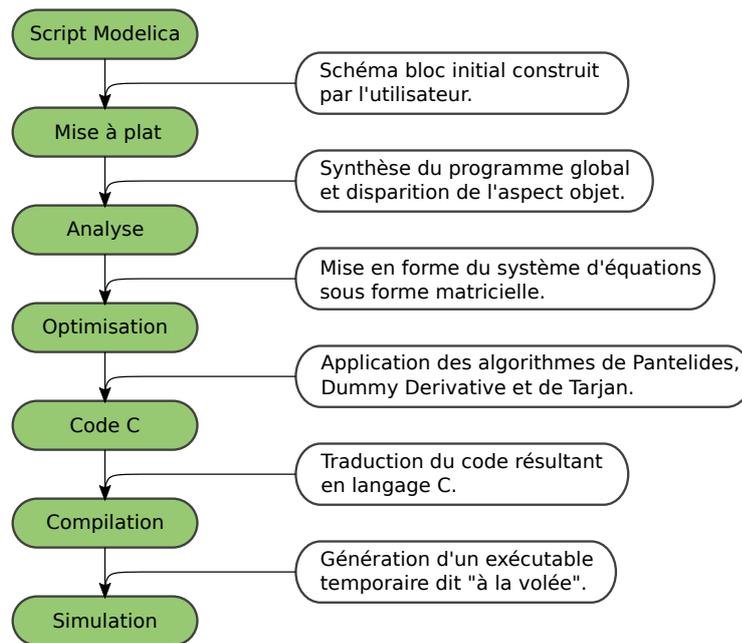


FIGURE B.1 – Résolution d'un système d'équations algébro-différentiel par OpenModelica.

C

Annexe - Chapitre 2

C.1 Code Python pour le calcul des indices de Shapley

```
1 import numpy as np
2
3 #Inputs:
4 # n: number of samples
5 # d: number of independent input variables
6 # x_LB (numpy.array, size [d,1]): lower bound of the array of input
   variables
7 # x_UB (numpy.array, size [d,1]): upper bound of the array of input
   variables
8 # myfunc (function handle): the user-defined function
9 #Outputs:
10 # phi1 (numpy.array, size [d,1]): the array collecting the Shapley's
   indices
11 # phi2 (numpy.array, size [d,1]): the array collecting the variance
   of Shapley's indices
12 # var_tot: the total variance of the output
13 def Shapley(n,d,x_LB,x_UB,myfunc):
14     x = np.zeros((n,d)) #initialise array x
15     y = np.zeros((n,d)) #initialise array y
16
17     for i in range(d):
18         # fill array x with n samples uniformly distributed between x_LB
           and x_UB along each dimension
19         x[:,i] = np.random.uniform(x_LB[i], x_UB[i], (1,n))
20         # fill array y with n samples uniformly distributed between x_LB
           and x_UB along each dimension
21         y[:,i] = np.random.uniform(x_LB[i], x_UB[i], (1,n))
22
23     shu = np.arange(0, d, 1)
24     pm = np.zeros((n,d))
25
```

```

26 for i in range(n):
27     pm[i] = np.take(shu, np.random.permutation(shu.shape[0]), axis
28                 =0, out=shu)
29
30 a = np.arange(0, d, 1)
31
32 z = x.copy()
33 fz1 = myfunc(z)
34 fx = fz1
35
36 phi1 = np.zeros(d) # initialization
37 phi2 = np.zeros(d) # initialization
38
39 ind = np.zeros((n,d)).astype(bool)
40
41 for j in range(d):
42     for k in range(n):
43         ind[k] = (pm[k, j] == a)
44         y_red = y*ind
45
46         for i1 in range(n):
47             for i2 in range(d):
48                 if(y_red[i1, i2] != 0):
49                     z[i1, i2] = y_red[i1, i2]
50
51         fz2 = myfunc(z)
52         fmarg = ((fx-fz1/2-fz2/2)*(fz1-fz2))
53
54         #Shapley's index for input variable j
55         phi1 = phi1 + fmarg @ (ind/n)
56
57         #Variance of Shapley's index
58         phi2 = phi2 + fmarg**2 @ (ind/n)
59         fz1 = fz2
60
61 var_tot = sum(phi1) #total variance of the output
62
63 return phi1, phi2, var_tot

```

D

Annexe - Chapitre 4

D.1 Plaque percée : phénomène d'*overfitting*

Cette section reprend le cas de la plaque percée en flexion, de la section 4.3.2 du chapitre 4, sans considérer le terme de lissage dans la génération du métamodèle. Cela a pour but d'illustrer le phénomène d'*overfitting*. Pour rappel, le métamodèle possède $M = 1$ sortie, u_z et $N = 4$ variables d'entrée, les coordonnées x et y , l'épaisseur de la plaque e_{plq} et la force F . Les caractéristiques du métamodèle sont rappelées dans le Tab. D.1.

TABLE D.1 – Caractéristiques du métamodèle appliqué sur le problème de la plaque percée.

Variable	Intervalle	p_k	r_k	n_k	n_{TP}	n_{CP}	β_i	ω_{i_1, \dots, i_N}
x	$\mathcal{U}[0, 200]$		28	27				
y	$\mathcal{U}[0, 100]$		13	12				
F	$\mathcal{U}[150, 300]$	2	12	8	68614	29484	$]0, 1[$	$]1, 10[$
e_{plq}	$\mathcal{U}[1, 10]$		12	8				
\mathbf{Q}_c			(31, 16, 6, 6)			26656		

Les valeurs des indicateurs de précision, R_α^2 et $\bar{\varepsilon}$, évalués sur une base de données \mathbf{Q}_c , ainsi que les valeurs de la fonction objectif f_{obj} , sont renseignées dans le Tab. D.2. Bien que la fonction objectif diminue tout au long du processus de génération du métamodèle, traduisant ainsi une bonne optimisation, les valeurs des autres indicateurs se détériorent. Dans cette situation, cela est dû à un phénomène d'*overfitting*, illustré par la vue en coupe du champ de déplacement de la plaque percée dans la Fig. D.1 : du bruit dans le comportement du métamodèle apparaît et fausse les résultats.

TABLE D.2 – Évolution des coefficients f_{obj} , R_α^2 et $\bar{\varepsilon}_1$, en fonction de l'enrichissement de la base de données d'entraînement et de l'étape d'optimisation considérée, sans prise en compte du terme de lissage.

f_{obj}			R_α^2			$\bar{\varepsilon}_\alpha$ [%]		
$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$	$\xi_{1,\text{opt}}$	$\xi_{2,\text{opt}}$	$\xi_{3,\text{opt}}$
35.0200	0.7100	0.0246	0.997	0.994	0.994	1.54	2.28	2.29

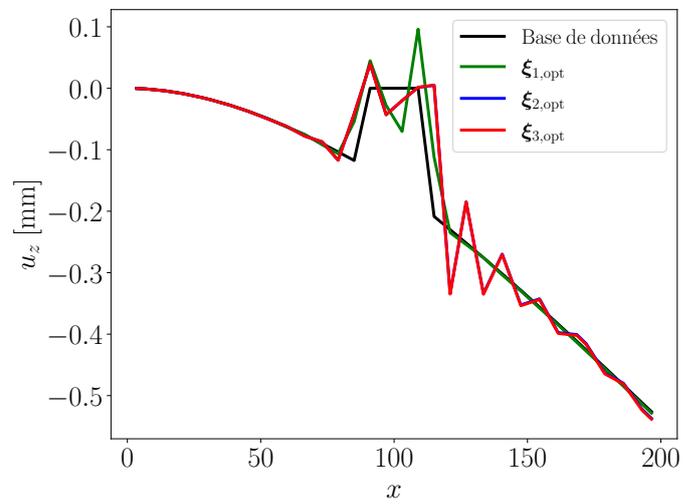


FIGURE D.1 – Vue en coupe de la plaque percée pour $F = 158$ N, $e_{plq} = 4.52$ mm et $y = 56$ mm en fonction de l'étape de génération du métamodèle, sans prise en compte du terme de lissage.

E

Annexe - Chapitre 5

E.1 Modèles analytiques thermiques

Hormis la mullite, les comportements des différentes couches thermiques de la Fig. E.1, sont décrits analytiquement par quatre lois classiques de la thermique, rappelées dans [95]. La première relation exprime le flux reçu par un matériau en fonction de sa capacité thermique, en 0-D :

$$\phi_{\text{capa}} = mc_p \frac{\partial T}{\partial t} \quad (\text{E.1})$$

avec ϕ_{capa} le flux de chaleur reçu (positif) ou émis (négatif) par le matériau, m la masse de matériau, c_p sa capacité thermique massique à pression constante, et T la température du matériau.

La seconde relation exprime le flux au travers ce matériau en fonction de sa conductivité, en 0-D :

$$\phi_{\text{cond}} = \frac{\lambda A}{e} \Delta T \quad (\text{E.2})$$

Avec ϕ_{cond} le flux de chaleur à travers le matériau, $A = \pi R^2$ la surface de sa section, normale au flux de chaleur, e son épaisseur, λ sa conductivité thermique et ΔT la différence de température entre les deux points de mesure du flux (sans changement de matière).

La troisième relation exprime le flux thermique entre un fluide et une paroi. Lorsque le fluide circule en contact de la paroi, on parle de convection naturelle. Lorsque la circulation du fluide est générée par une source mécanique, on parle de convection forcée. Dans les deux cas, le flux convectif 0-D est exprimé par :

$$\phi_{\text{conv}} = h_{\text{conv}} A \Delta T \quad (\text{E.3})$$

avec ϕ_{conv} le flux de chaleur entre la paroi et le fluide, h le coefficient de convection, A la surface d'échange, et $\Delta T = T_p - T_f$ la différence de température entre la paroi T_p et le fluide T_f . Toute la difficulté est de parvenir à exprimer le coefficient de convection h_{conv} , très dépendant de la plage de température de travail, de la géométrie étudiée, du régime de circulation du fluide et du fluide lui-même. Dans la situation du système embarqué, la convection est naturelle et la géométrie, associée aux couches d'air, est assimilable à deux

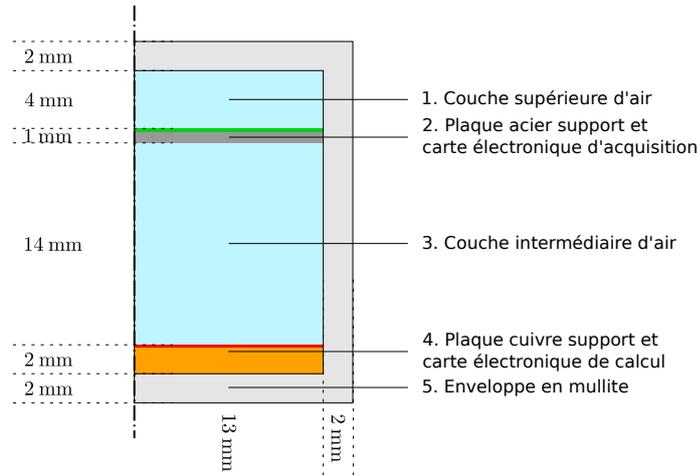


FIGURE E.1 – Schéma du Système Embarqué à Température Contrôlée. Il se compose, dans sa partie inférieure, d'une plaque supportant la carte électronique de calcul et, dans sa partie supérieure, d'une carte électronique d'acquisition. Les autres couches thermiques correspondent à des couches isolantes d'air. L'ensemble est entouré d'une enveloppe en mullite.

plaques en vis à vis séparées par une couche d'air. La plage de travail de température est de l'ordre de 100 K, entre 293 K et 393 K. En considérant les annexes de [95], il est possible de calculer une bonne approximation de ce coefficient convectif. La première étape est de déterminer les nombres de Prandtl [95] et Grashof [95] valant respectivement 10^9 et 10^4 dans ce contexte. Cela traduit un écoulement laminaire de l'air entre ces parois séparées par une distance de l'ordre de 10^{-2} m. Ainsi le coefficient de convection h_{conv} peut s'exprimer par :

$$h_{\text{conv}} = 1.32 \left(\frac{\Delta T}{L} \right)^{\frac{1}{4}} \iff h_{\text{conv}} \approx h_{\text{conv,MCP/air}} \approx h_{\text{conv,air/acier}} \approx 7. \quad (\text{E.4})$$

Cette valeur est cohérente avec la littérature [95], étant donné que pour l'air en convection naturelle, le coefficient convectif est compris entre 5 et 15.

La dernière loi thermique exprime le flux thermique radiatif en fonction des états de surface des parois considérées. Un flux radiatif émis par une paroi est dépendant de la géométrie de cette paroi, de son environnement, du matériau et de son émissivité. La forme générale d'un flux radiatif est donnée par :

$$\Phi_{\text{rad}} = \epsilon A \sigma T^4 \quad (\text{E.5})$$

avec ϵ l'émissivité de l'objet, A sa surface d'échange, $\sigma = 5.67 \cdot 10^{-8} \text{ W m}^{-2} \text{ K}^{-4}$ la constante de Stefan-Boltzmann et T la température de la paroi. Dans le cas d'étude de ce chapitre, on considère que le flux radiatif dans les cavités correspond au flux échangé par deux

parois en vis à vis de même section. Ainsi, l'Eq. E.5 devient :

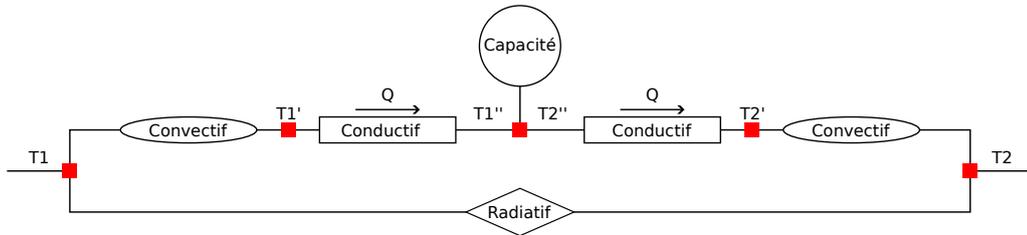
$$\Phi_{\text{rad}} = \frac{1}{\frac{1}{\epsilon_1} + \frac{1}{\epsilon_2} - 1} A\sigma(T_1^4 - T_2^4), \quad (\text{E.6})$$

avec ϵ_1 et T_1 respectivement l'émissivité et la température de la première paroi et ϵ_2 et T_2 celles de la seconde paroi.

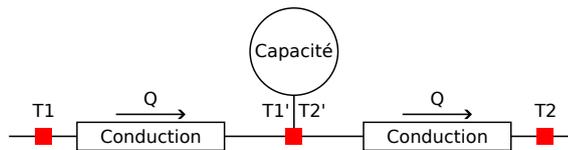
E.2 Sous-systèmes MBSE globaux

Les deux principaux sous-systèmes globaux employés dans un modèle thermique sont :

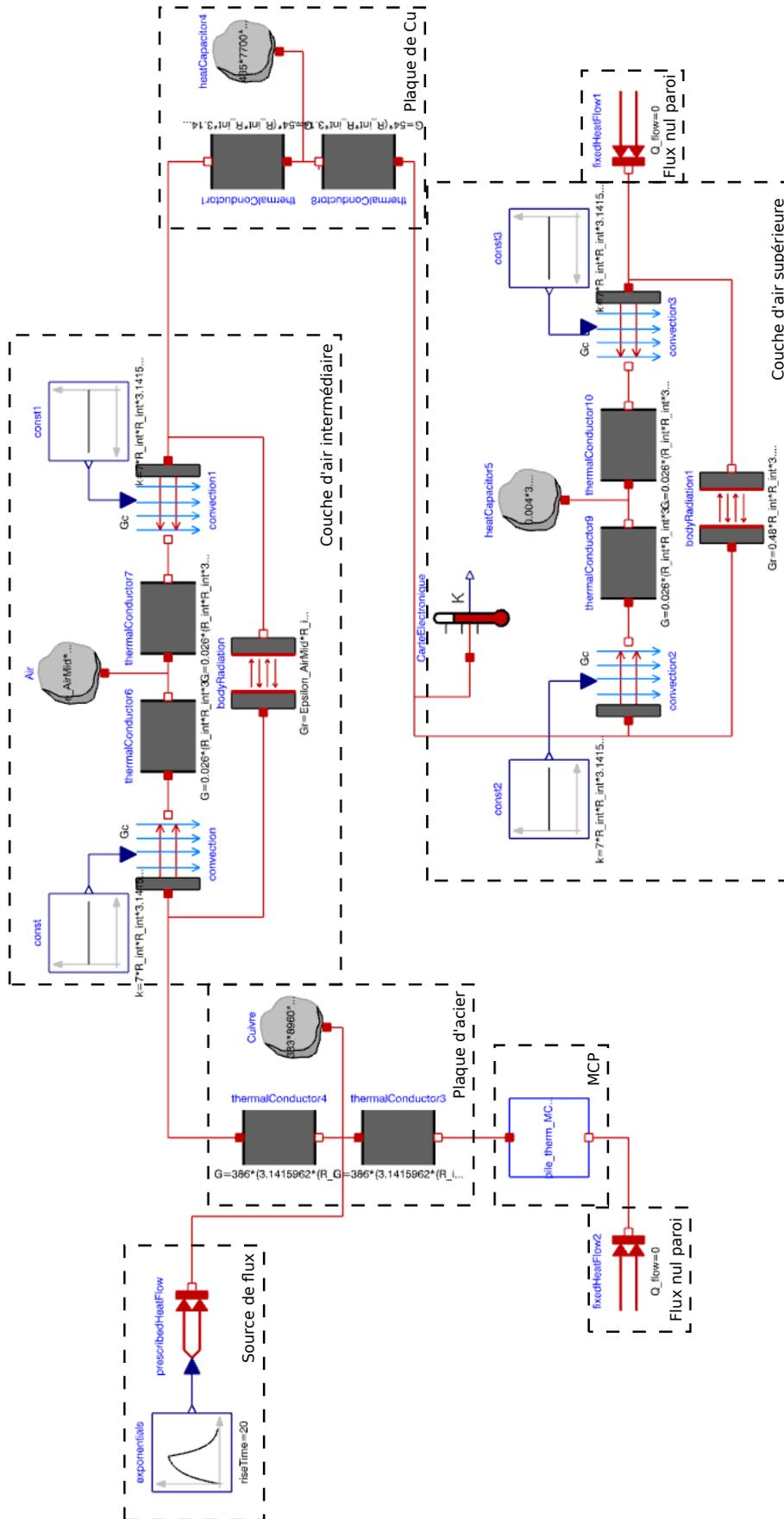
- les modèles de cavités remplies d'un fluide composées d'un élément capacitif en série, en haut et en bas, avec un demi élément conductif lui-même en série avec un élément convectif. Un élément radiatif est relié à l'ensemble en parallèle. Cet assemblage est représenté par :



- les modèles de parois composés d'un élément capacitif central en série au dessus et en dessous avec un demi élément conductif. Pour les parois avec une source, cette dernière est directement appliquée à la formulation de l'élément capacitif. Cet assemblage est représenté par :



E.3 Illustration du système thermique MBSE complet pour le cas SETC-2



E.4 Étude de convergence

Cette section reporte l'étude de convergence réalisée sur le maillage du modèle EF employé pour modéliser la partie inférieure du système embarqué. Les observables sont la variation de l'énergie interne totale du MCP, notée ΔE_{MCP} , et la contrainte maximale dans la mullite, notée $\sigma_{\text{max,Mu}}$. Les courbes de convergence de la Fig. E.2 tracent l'évolution des erreurs relatives de la valeur de l'observable à une taille de maille donnée par rapport à la valeur calculée avec la taille de maille précédente. La convergence est considérée atteinte lorsque cette erreur relative est inférieure à 5 %. Dans le cas de l'évolution de l'énergie interne, cela correspond à une taille de maille de $7.5 \cdot 10^{-3}$ m. Dans le cas de l'évolution de la contrainte dans la mullite, la convergence est atteinte pour une taille de maille de $5 \cdot 10^{-3}$ m. Néanmoins pour cette dernière, l'erreur augmente pour des tailles de maille inférieures à $2.5 \cdot 10^{-3}$ m. Cette augmentation est provoquée par le fait que la zone négligée autour de l'angle inférieur droit est trop restreinte : des valeurs non désirées sont prises en compte dans l'analyse. En considérant ces observations, le choix est donc fait de considérer une taille de maille moyenne de $2.5 \cdot 10^{-3}$ m, soit 64×68 éléments équivalent à 65×69 nœuds.

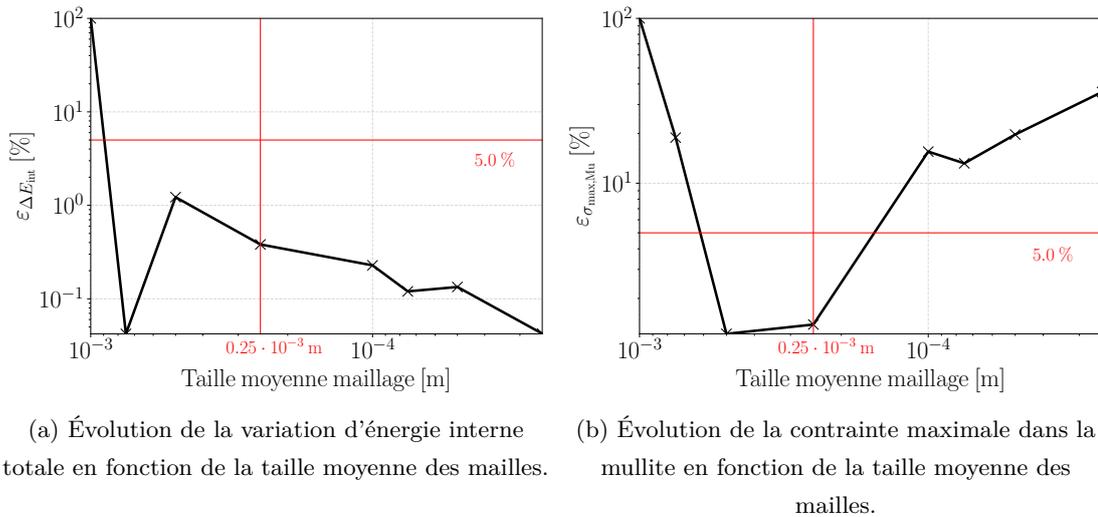


FIGURE E.2 – Courbes de convergence : diminution de l'erreur sur calcul de $\sigma_{\text{max,Mu}}$ et de ΔE_{int} entre deux maillages successifs.

E.5 Article publié dans une revue internationale, ou en préparation

(En préparation) Vuillod, B., Montemurro, M., Panettieri, E., and Hallo, L. (2024). Hybrid modeling strategy using an MBSE approach including NURBS-based metamodels applied to a thermomechanical problem.

(En préparation) Vuillod, B., Montemurro, M., Panettieri, E., and Hallo, L. (2024). On the generation of a metamodel based on NURBS entities as a solution of an optimisation problem.

Vuillod, B., Montemurro, M., Panettieri, E., and Hallo, L. (2023). A comparison between Sobol's indices and Shapley's effect for global sensitivity analysis of systems with independent input variables. *Reliability Engineering and System Safety*, 234(February), 109177. <https://doi.org/10.1016/j.ress.2023.109177>

E.6 Article publié dans un *proceeding* de conférence internationale

Vuillod, B., Zani, M., Hallo, L., and Panettieri, E. (2023). A surrogate model based on NURBS entities for engineering problems. *Proceedings of the 6th international conference on soft computing, machine learning and optimisation in civil, structural and environmental engineering*, 1–12. <https://doi.org/10.4203/ccc.5>

Vuillod, B., Panettieri, E., Hallo, L., and Montemurro, M. (2023). Preliminary Design of Non-Linear Systems Based on Global Sensitivity Analysis and Modelica Language. *Proceedings of the Design Society*, 3(July), 2925–2934. <https://doi.org/10.1017/pds.2023.293>

Chevalier, R., Vuillod, B., Zani, M., Montemurro, M., Pommier, R., and Catapano, A. (2023). Numerical Homogenisation of Mechanical Properties of Pinus Pinaster (Ait.) Lamellae Constituting Glued Laminated Timber. *13th World Conference on Timber Engineering, WCTE 2023*, 1, 455–464. <https://doi.org/10.52202/069179-0062>

Vuillod, B., Hallo, L., Panettieri, E., and Montemurro, M. (2021). Sensitivity Analysis of a Car Shock Absorber Through a Functional Mock-up Units-Based Modelling Strategy. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*,

307-314. <https://doi.org/10.3384/ecp21181307>

Bibliographie

- [1] A. Ammar. Intelligence artificielle, jumeaux digitaux et réduction dimensionnelle, 2022. (Cité page 30)
- [2] A. Ammar, A. Huerta, F. Chinesta, E. Cueto, and A. Leygue. Parametric solutions involving geometry : A step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268 :178–193, jan 2014. (Cité page 36)
- [3] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. Part II : Transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics*, 144(2-3) :98–121, 2007. (Cité pages 36 et 139)
- [4] R. Andrzejczyk, M. Saqib, T. Kowalczyk, and H. M. Ali. Study on effective front region thickness of PCM in thermal energy storage using a novel semi-theoretical model. *International Communications in Heat and Mass Transfer*, 146(June) :106901, 2023. (Cité page 113)
- [5] D. André, I. Iordanoff, J. Charles, and J. Néauport. Discrete element method to simulate continuous material by using the cohesive beam model. *Journal of Computer Methods in Applied Mechanics and Engineering*, 213-216 :113–125, 2012. (Cité page 18)
- [6] D. André, J. luc Charles, I. Iordanoff, and J. Néauport. The granoo workbench, a new tool for developing discrete element simulations, and its application to tribological problems. *Advances in Engineering Software*, 74 :40–48, 2014. (Cité page 18)
- [7] Ansys. ANSYS Mechanical APDL Element Reference. *Knowledge Creation Diffusion Utilization*, (October) :1–1416, 2012. (Cité pages 136, 137, et 144)
- [8] A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs*. Oxford Statistical Science Series, 1992. (Cité page 27)
- [9] Y. Audoux. *Développement d’une nouvelle méthode de réduction de modèle basée sur les hypersurfaces NURBS (Non-Uniform Rational B-Splines) Yohann Audoux To cite this version : HAL Id : tel-02463073*. PhD thesis, ED SMI 432, 2020. (Cité pages 5, 6, 43, 46, 48, 85, 97, 108, et 123)

-
- [10] Y. Audoux, M. Montemurro, and J. Pailhès. A Metamodel Based on Non-Uniform Rational Basis Spline Hyper-Surfaces for Optimisation of Composite Structures. *Composite Structures*, 247, 2020. (Cité pages 5, 43, 46, 48, 85, et 86)
- [11] Y. Audoux, M. Montemurro, and J. Pailhès. Non-Uniform Rational Basis Spline hyper-surfaces for metamodeling. *Computer Methods in Applied Mechanics and Engineering*, 364 :112918, 2020. (Cité pages 5, 43, 46, 48, 85, 86, 92, 93, 95, 97, 99, et 139)
- [12] S. Baillargeon. Le Krigeage : revue de la théorie et application à l'interpolation spatiale de données de précipitations. *Faculté des sciences et de génie*, Maîtrise :137, 2005. (Cité pages 39, 40, et 41)
- [13] H. T. Banks, M. L. Joyner, B. Wincheski, and W. P. Winfree. Nondestructive evaluation using a reduced-order computational methodology. *Inverse Problems*, 16(4) :929, aug 2000. (Cité page 34)
- [14] M. Baudin, A. Dutfoy, B. Iooss, and A.-L. Popelin. *OpenTURN : An Industrial Software for Uncertainty Quantification in Simulation*, pages 1–38. Springer International Publishing, Cham, 2016. (Cité page 41)
- [15] W. Becker. Metafunctions for benchmarking in sensitivity analysis. *Reliability Engineering & System Safety*, 204 :107189, 12 2020. (Cité page 58)
- [16] N. Benoumechiara and K. Elie-Dit-Cosaque. Shapley effects for sensitivity analysis with dependent inputs : bootstrap and kriging-based algorithms. *ESAIM : Proceedings and Surveys*, 65 :266–293, 2019. (Cité pages 58, 60, et 61)
- [17] G. Bertolino, M. Montemurro, and G. De Pasquale. Multi-scale shape optimisation of lattice structures : an evolutionary-based approach. *International Journal on Interactive Design and Manufacturing*, 13(4) :1565–1578, 2019. (Cité pages 5, 43, 46, 48, et 86)
- [18] G. Bertolino, M. Montemurro, N. Perry, and F. Pourroy. An Efficient Hybrid Optimization Strategy for Surface Reconstruction. *Computer Graphics Forum*, 40(6) :215–241, 2021. (Cité pages 5, 6, 85, 86, 90, 91, 92, 93, 95, 96, et 139)
- [19] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauss, H. Elmquist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel. Functional Mockup Interface 2.0 : The Standard for Tool independent Exchange of Simulation Models. In *Proceedings of the 9th International Modelica Conference, September 3-5, 2012, Munich, Germany*, volume 76, pages 173–184, 2012. (Cité page 22)
- [20] B. Bognet, F. Chinesta, A. Leygue, and A. Poitou. Proper Generalized Decomposition (PGD) et séparation de variables spatiales pour la résolution en thermoélasticité linéaire appliquée à des plaques composites. *CSMA2011*, pages 1–8, 2011. (Cité page 36)

- [21] P.-A. Boucar, M. Derumaux, P. Ladevèze, and P. Roux. Macro-meso models for joint submitted to pyrotechnic shock. *Computational Fluid and Solid Mechanics*, pages 139–142, 2003. (Cité page 36)
- [22] B. Broto, F. Bachoc, M. Depecker, and J. M. Martinez. Sensitivity indices for independent groups of variables. *Mathematics and Computers in Simulation*, 163, 2019. (Cité page 57)
- [23] C. G. Bucher and U. Bourgund. A fast and efficient response surface approach for structural reliability problems. *Structural Safety*, 7(1) :57–66, 1990. (Cité pages 123, 127, et 128)
- [24] K. R. Butterfield. The computation of all the derivatives of a b-spline basis. *IMA Journal of Applied Mathematics (Institute of Mathematics and Its Applications)*, 17(1) :15–25, 1976. (Cité page 107)
- [25] L. F. Cabeza and N. H. S. Tay. *High-Temperature Thermal Storage Systmes Using Phase Change Materials*. Elsevier Inc., 1st edition, 2017. (Cité page 158)
- [26] P. Campanini and G. Ferretti. Object-Oriented Models of Parallel Manipulators. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden*, 181 :241–248, 2021. (Cité page 19)
- [27] E. Carrera, G. A. Fiordilino, M. Nagaraj, A. Pagani, and M. Montemurro. A global/local approach based on CUF for the accurate and efficient analysis of metallic and composite structures. *Engineering Structures*, 188(November 2018) :188–201, 2019. (Cité page 47)
- [28] A. Catapano and M. Montemurro. Strength optimisation of variable angle-tow composites through a laminate-level failure criterion. *Journal of Optimization Theory and Applications*, 181 :683–706, 2020. (Cité page 86)
- [29] L. Chamoin, C. Jailin, M. Diaz, and L. Quesada. Coupling between topology optimization and digital image correlation for the design of specimen dedicated to selected material parameters identification. *International Journal of Solids and Structures*, 193-194 :270–286, 2020. (Cité page 43)
- [30] G. Chastaing, F. Gamboa, and C. Prieur. Generalized Hoeffding-Sobol decomposition for dependent variables - application to sensitivity analysis. *Electronic Journal of Statistics*, 6, 2012. (Cité pages 26 et 57)
- [31] G. Chastaing, F. Gamboa, and C. Prieur. Generalized Sobol sensitivity indices for dependent variables : numerical methods. *Journal of Statistical Computation and Simulation*, 85(7) :1306–1333, 2015. (Cité page 26)
- [32] A. Chatterjee. An introduction to the proper orthogonal decomposition on JSTOR. *Current Science*, 78(7) :808–817, 2000. (Cité page 34)

-
- [33] J.-C. Chaudemar and P. De Saqui-Sannes. MBSE and MDAO for Early Validation of Design Decisions : a Bibliography Survey. In *15th Annual IEEE International Systems Conference, SysCon 2021*, volume 15, page 9, Toulouse, 2021. (Cité page 15)
- [34] F. Chinesta, R. Keunings, and A. Leygue. *The proper generalized decomposition for advanced numerical simulations : A primer*. Number 9783319028644. 2014. (Cité pages 123, 124, 125, et 126)
- [35] F. Chinesta and P. Ladevèze. Proper generalized decomposition. In *Volume 2 Snapshot-Based Methods and Algorithms*, chapter 3, pages 97–138. De Gruyter, 2021. (Cité page 37)
- [36] J. Y. Cognard and P. Ladevèze. A large time increment approach for cyclic viscoplasticity. *International Journal of Plasticity*, 9(2) :141–157, jan 1993. (Cité page 36)
- [37] P. G. Constantine. *Active Subspaces*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2015. (Cité page 26)
- [38] G. Costa, M. Montemurro, and J. Pailhès. Minimum length scale control in a nurbs-based simp method. *Computer Methods in Applied Mechanics and Engineering*, 354 :963–989, 2019. (Cité pages 5, 6, 43, 48, 86, 90, 91, 92, 93, 111, 113, et 116)
- [39] G. Costa, M. Montemurro, and J. Pailhès. Nurbs hyper-surfaces for 3d topology optimization problems. *Mechanics of Advanced Materials and Structures*, 28(7) :665–684, 2021. (Cité pages 5, 43, 48, 86, et 128)
- [40] M. G. Cox. The Numerical Evaluation of B-Splines*. *IMA Journal of Applied Mathematics*, 10(2) :134–149, oct 1972. (Cité page 43)
- [41] N. A. C. Cressie. *Statistics for spatial data*. John Wiley and Sons Inc., New York, 2 edition, 1993. (Cité page 41)
- [42] R. I. Cukier et al. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. Theory. *The Journal of Chemical Physics*, 59(8), 1973. (Cité pages 29, 57, 59, et 61)
- [43] R. I. Cukier, H. B. Levine, and K. E. Shuler. Nonlinear sensitivity analysis of multiparameter model systems. *Journal of Computational Physics*, 26(1) :1–42, jan 1978. (Cité page 57)
- [44] S. Da Veiga, F. Gamboa, B. Iooss, and C. Prieur. *Basics and Trends in Sensitivity Analysis Theory and Practice in R*. Society for Industrial and Applied Mathematics, U.S., 2021. (Cité pages 54 et 55)
- [45] Dassault Systèmes AB. 3D Experience : User’s Guide, 2023. (Cité page 19)
- [46] C. De Boor. *A Practical Guide to Splines.*, volume 34. 1980. (Cité pages 38 et 44)

- [47] R. Demachy and S. Guilmeau. Short paper - Structural consistency of MBSE and MBSA models using Consistency Links. In *11th European Congress Embedded Real Time System*, volume 1, page 5, Toulouse, 2022. (Cité page 15)
- [48] X. Du and L. Leifsson. Multifidelity modeling by polynomial chaos-based cokriging to enable efficient model-based reliability analysis of ndt systems. *Journal of Nondestructive Evaluation*, 38, 2020. (Cité page 38)
- [49] M. Eklund, J. Savolainen, A. Lukkari, and T. Karhela. Optimizing life-cycle costs for pumps and powertrains using FMI co-simulation. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*, 181 :681–689, 2021. (Cité page 23)
- [50] B. El Hefni and D. Bouskela. *Modeling and Simulation of Thermal Power Plants with ThermoSysPro*. Springer Cham, 2019. (Cité page 16)
- [51] K.-T. Fang, R. Li, and A. Sudjianto. *Design and Modeling for Computer Experiments*. Capman and Hall/CRC, New York, 1st editio edition, 2005. (Cité pages 26 et 27)
- [52] G. Farin. *Curves and surfaces for CAGD : a practical guide*. Morgan Kaufmann, 5 edition, 2002. (Cité page 85)
- [53] G. A. Fiordilino, M. I. Izzì, and M. Montemurro. A general isogeometric polar approach for the optimisation of variable stiffness composites : application to eigenvalue buckling problems. *Mechanics of Materials*, 153, 2021. (Cité page 86)
- [54] A. S. Fleischer. *Energy Storage Applications*, pages 7–35. Springer International Publishing, Cham, 2015. (Cité page 158)
- [55] A. S. Fleischer. *PCM Design Issues*, pages 49–73. Springer International Publishing, Cham, 2015. (Cité pages 158 et 160)
- [56] M. S. Floater. Meshless Parameterization and B-Spline Surface Approximation. *The Mathematics of Surfaces IX*, pages 1–18, 2000. (Cité pages 5, 85, et 95)
- [57] K. Forsberg and H. Mooz. The Relationship of System Engineering to the Project Cycle. In *INCOSE International Symposium*, pages 57 – 65, 1991. (Cité page 16)
- [58] V. Fournier. Des revêtements autoportés en aluminosilicates jusqu’à 1 cm d’épaisseur et 80 plasma : développement et caractérisation thermomécanique To cite this version : HAL Id : tel-03334456 Des revêtements autoportés en aluminosili. 2021. (Cité pages 161, 166, et 171)
- [59] P. Fritzson. *Principles of Object Oriented Modeling and Simulation with Modelica 3.3 : A Cyber-Physical Approach*. IEEE Press, 2014. (Cité pages 19, 21, 64, et 190)
- [60] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. 2 edition, 1990. (Cité page 34)

-
- [61] A. Gálvez and A. Iglesias. From nonlinear optimization to convex optimization through firefly algorithm and indirect approach with applications to CAD/CAM. *The Scientific World Journal*, 2013. (Cité pages 91 et 92)
- [62] D. Garcia, I. Arostegui, and R. Prellezo. Robust combination of the morris and sobol methods in complex multidimensional models. *Environmental Modelling & Software*, 122 :14, 2019. (Cité page 28)
- [63] J. C. García-Merino, C. Calvo-Jurado, E. Martínez-Pañeda, and E. García-Macías. Multielement polynomial chaos Kriging-based metamodelling for Bayesian inference of non-smooth systems. *Applied Mathematical Modelling*, 116(December) :510–531, 2023. (Cité page 132)
- [64] J. Gauchi. *Plans d'expériences optimaux pour modèles linéaires*. Paris, 1997. (Cité page 27)
- [65] C. Ghnatios, E. Cueto, A. Falco, J.-L. Duval, and F. Chinesta. Spurious-free interpolations for non-intrusive PGD-based parametric solutions : Application to composites forming processes. *International Journal of Material Forming*, 14(1) :83–95, 2020. (Cité page 37)
- [66] T. Goda. A simple algorithm for global sensitivity analysis with Shapley effects. *Reliability Engineering & System Safety*, page 107702, 2021. (Cité pages 53, 59, 60, 61, et 66)
- [67] J. Goffart and M. Woloszyn. RBD-FAST : une méthode d'analyse de sensibilité rapide et rigoureuse pour la garantie de performance énergétique. *Conférence Francophone de l'International Building Performance Simulation Association IBPSA,,* pages 1–8, 2018. (Cité pages 29 et 57)
- [68] C. Gomes, T. Blochwitz, C. Bertsch, K. Wernersson, K. Schuch, P. R., O. Kotte, I. Zacharias, M. Blesken, T. Sommer, M. Najafi, and A. Junghanns. The FMI 3.0 Standard Interface for Clocked and Scheduled Simulations. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*, 181 :27–36, 2021. (Cité page 22)
- [69] P. Goovaerts. *Geostatistics for natural resources evaluation*. Oxford University Press, 1997. (Cité page 41)
- [70] J. Goupy. *Plans d'expériences pour surface de réponse*. 1999. (Cité pages 26 et 27)
- [71] K. Gurney. *An Introduction to Neural Networks*. Taylor and Francis, Inc., USA, 1997. (Cité pages 42 et 43)
- [72] C. Haberstich and C. Cannamela. Cours workshop : Plans d'expériences pour métamodèles, 2022. (Cité pages 25, 27, 97, et 128)
- [73] A. Halimov, M. Lauster, and D. Müller. Validation and integration of a latent heat storage model into building envelopes of a high-order building model for Modelica library AixLib. *Energy and Buildings*, 2019. (Cité page 159)

- [74] J. H. Halton. Radical-inverse quasi-random point sequence. *Communication of the ACM*, 7 :701–702, 1964. (Cité page 27)
- [75] S. U. Hamim and R. P. Singh. Proper Orthogonal Decomposition-Radial Basis Function Surrogate Model-Based Inverse Analysis for Identifying Nonlinear Burgers Model Parameters from Nanoindentation Data. *Journal of Engineering Materials and Technology, Transactions of the ASME*, 139(4), oct 2017. (Cité page 34)
- [76] N. Hasmukhbhai Shah, P. Le Henaff, C. Schiffer, M. Krammer, and M. Benedikt. Accurate Robot Simulation for Industrial Manufacturing Processes using FMI and DCP Standards. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*, 181 :673–679, 2021. (Cité page 23)
- [77] S. Haykin. Neural networks : a comprehensive foundation by Simon Haykin. *The Knowledge Engineering Review*, 13(4) :409–412, 1999. (Cité pages 42 et 43)
- [78] J. He and Z.-F. Fu. Mathematics for modal analysis. In *Modal Analysis*, chapter 2, pages 12–48. Butterworth-Heinemann, jan 2001. (Cité page 31)
- [79] M. B. Heredia, C. Prieur, and N. Eckert. Global sensitivity analysis with aggregated Shapley effects, application to avalanche hazard assessment. *Reliability Engineering & System Safety*, 222 :245–251, 2022. (Cité page 58)
- [80] M. Hinze and S. Volkwein. Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems : Error Estimates and Suboptimal Control. *Dimension Reduction of Large-Scale Systems*, pages 261–306, oct 2005. (Cité page 34)
- [81] W. Hoeffding. A Class of Statistics with Asymptotically Normal Distribution. *Annals of Mathematical Statistics*, 19(3) :293–325, 1948. (Cité pages 24, 28, 54, et 55)
- [82] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. Turbulence, Coherent Structures, Dynamical Systems and Symmetry. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2012. (Cité page 34)
- [83] G. Hooker. Generalized functional ANOVA diagnostics for high-dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3), 2007. (Cité page 26)
- [84] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41) :4135–4195, 2005. (Cité page 86)
- [85] Z. Huimin, L. Yangyang, W. Li, D. Shuhong, W. Jun, C. Liping, Z. Fanli, D. Ji, and Z. Haiming. Modelica Modeling and Simulation for a Micro Gas-Cooled Reactor. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*, 181 :569–575, 2021. (Cité page 19)
- [86] M. Imran Khan, F. Asfand, and S. G. Al-Ghamdi. Progress in research and development of phase change materials for thermal energy storage in concentrated solar power. *Applied Thermal Engineering*, 219(PB) :119546, 2023. (Cité pages 158 et 160)

-
- [87] B. Iooss. Analyses de sensibilité globales de modèles complexes, quelques applications dans le nucléaire. 2006. (Cité page 28)
- [88] B. Iooss. Revue sur l'analyse de sensibilité globale de modèles numériques. *Journal de la Société Française de Statistiques*, 152(1) :3–25, 2010. (Cité pages 25, 26, 162, et 172)
- [89] B. Iooss and P. Lemaître. A review on global sensitivity analysis methods. *Uncertainty management in simulation-optimization of complex system.*, 59 :101–122, 2015. (Cité pages 24, 25, 55, et 57)
- [90] B. Iooss and C. Prieur. Shapley effects for sensitivity analysis with correlated inputs : Comparisons with Sobol' indices, numerical estimation and applications. *International Journal for Uncertainty Quantification*, 9(5) :493–514, 2019. (Cité pages 57 et 58)
- [91] T. Ishigami and T. Homma. An Importance Quantification Technique in Uncertainty Analysis for Computer Models. In *Proceedings. First International Symposium on Uncertainty Modeling and Analysis*, volume 21, pages 99–104, College Park, MD, USA, 1990. (Cité page 58)
- [92] E. Iuliano and D. Quagliarella. Proper Orthogonal Decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. *Computers & Fluids*, 84 :327–350, sep 2013. (Cité page 35)
- [93] M. I. Izzi, A. Catapano, and M. Montemurro. Strength and mass optimisation of variable-stiffness composites in the polar parameters space. *Structural and Multidisciplinary Optimization*, 64, 2021. (Cité pages 76, 86, et 143)
- [94] M. I. Izzi, M. Montemurro, and A. Catapano. Variable-stiffness composites optimisation under multiple design requirements and loads. *International Journal of Mechanical Sciences*, 258, 2023. (Cité page 86)
- [95] Y. Jannot. Transferts thermiques. *Ecole des Mines Nancy*, 2012. (Cité pages 124, 155, 195, et 196)
- [96] I. Kalogridis. Robust thin-plate splines for multivariate spatial smoothing. *Econometrics and Statistics*, pages 1–13, 2023. (Cité page 94)
- [97] J. P. Kleijnen. Sensitivity analysis and optimization of system dynamics models : Regression analysis and statistical design of experiments. *System Dynamics Review*, 11(4) :275–288, 1995. (Cité page 25)
- [98] J. Koehkler and A. B. Owen. Computer Experiments. *Handbook of Statistics*, 13 :261–308, 1996. (Cité page 26)
- [99] K. Konakli and B. Sudret. Global sensitivity analysis using low-rank tensor approximations. *Reliability Engineering & System Safety*, 156 :64–83, 2016. (Cité pages 54, 55, 56, et 57)

- [100] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln : Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. (Cité page 101)
- [101] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. *SIAM Journal*, 40(2) :492–515, 2006. (Cité page 34)
- [102] P. Ladevèze. New algorithms : mechanical framework and development (in french). Technical report, LMT-Cachan, 1985. (Cité page 36)
- [103] P. Ladevèze, J. Y. Cognard, and P. Talbot. A Non-incremental and Adaptive Computational Approach in Thermo-viscoplasticity. pages 281–291, 1999. (Cité page 36)
- [104] P. Ladevèze, J. C. Passieux, and D. Néron. The LATIN multiscale computational method and the Proper Generalized Decomposition. *Computer Methods in Applied Mechanics and Engineering*, 199(21-22) :1287–1296, apr 2010. (Cité page 36)
- [105] P. Ladevèze and O. Zienkiewicz. New advances in the LArge Time INcrement method. *New Advances in Computational Structural Mechanics*, pages 3–21, 1991. (Cité page 36)
- [106] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Chinesta, and F. Jacquemin. Routes for Efficient Computational Homogenization of Nonlinear Materials Using the Proper Generalized Decompositions. *Archives of Computational Methods in Engineering*, 17(4) :373–391, dec 2010. (Cité page 36)
- [107] O. Le Maître, H. Najm, R. Ghanem, and O. Knio. Multi-resolution analysis of wiener-type uncertainty propagation schemes. *Journal of Computational Physics*, 197 :502–531, 07 2004. (Cité page 38)
- [108] E. T. Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6) :363–370, 1989. (Cité pages 91, 113, et 125)
- [109] F. Legoll and L. Chamoin. Certified Computations with PGD Model Reduction in the MsFEM Framework. 336(2018) :2018, 2021. (Cité page 47)
- [110] E. H. Lehmann. On the history and use of some standard statistical models. *Institute of Mathematical Statistics*, 2 :114–126, 2008. (Cité page 39)
- [111] R. I. Leine, D. H. Van Campen, A. De Kraker, and L. Van Den Steen. Stick-Slip Vibrations Induced by Alternate Friction Models. *Nonlinear Dynamics*, 16 :41–54, 1998. (Cité pages 68 et 69)
- [112] H. Lemoussu, P. A. Boucard, and P. Ladevèze. A 3D shock computational strategy for real assembly and shock attenuator. *Advances in Engineering Software*, 33(7-10) :517–526, 2002. (Cité page 36)
- [113] J. Y. Lettvin, H. R. Maturanat, W. S. McCulloch, and W. H. Pitts|. What the Frog’s Eye Tells the Frog’s Brain *. 1959. (Cité page 41)

-
- [114] A. Leygue and E. Verron. A First Step Towards the Use of Proper General Decomposition Method for Structural Optimization. *Archives of Computational Methods in Engineering*, 17(4) :465–472, dec 2010. (Cité page 36)
- [115] C. Liu, L. Liu, K. Tan, L. Zhang, K. Tang, and X. Shi. Fabrication and characterization of porous cordierite ceramics prepared from ferrochromium slag. *Ceramics International*, 42(1, Part A) :734–742, 2016. (Cité pages 166 et 171)
- [116] F. Liu, P. Wei, C. Tang, P. Wang, and Z. Yue. Global sensitivity analysis for multivariate outputs based on multiple response Gaussian process model. *Reliability Engineering & System Safety*, 189 :287–298, 2020. (Cité page 54)
- [117] T. A. Mara. Extension of the RBD-FAST method to the computation of global sensitivity indices. *Reliability Engineering & System Safety*, 94(8) :1274–1281, 2009. (Cité page 57)
- [118] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8) :1246–1266, 1963. (Cité page 40)
- [119] G. Matheron and F. Blondel. *Traité de géostatistique appliquée*, volume 1. France, 1962. (Cité page 40)
- [120] S. E. Mattsson and G. Söderlind. Index Reduction in Differential-Algebraic Equations Using Dummy Derivatives. *SIAM Journal on Scientific Computing*, 14(3) :677–692, 1993. (Cité page 189)
- [121] K. Mehrotra, C. K. Mohan, and S. Ranka. *Elements of artificial neural networks*. MIT Press, Cambridge, 1996. (Cité page 42)
- [122] N. Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247) :335–341, 1949. (Cité page 27)
- [123] M. J. Mifsud, S. T. Shaw, and D. G. MacManus. A high-fidelity low-cost aerodynamic model using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 63(4) :468–494, jun 2009. (Cité page 34)
- [124] M. Montemurro. An extension of the polar method to the first-order shear deformation theory of laminates. *Composite Structures*, 127 :328–339, 2015. (Cité pages 75 et 143)
- [125] M. Montemurro. On the structural stiffness maximisation of anisotropic continua under inhomogeneous neumann-dirichlet boundary conditions. *Composite Structures*, 287(115289), 2022. (Cité pages 93 et 103)
- [126] M. Montemurro and A. Catapano. On the effective integration of manufacturability constraints within the multi-scale methodology for designing variable angle-tow laminates. *Composite Structures*, 161 :145–159, 02 2017. (Cité pages 5, 43, et 48)

- [127] M. Montemurro and A. Catapano. A general b-spline surfaces theoretical framework for optimisation of variable angle-tow laminates. *Composite Structures*, 209 :561–578, 2019. (Cit  pages 5, 43, 48, et 86)
- [128] M. Montemurro, A. Catapano, and D. Doroszewski. A multi-scale approach for the simultaneous shape and material optimisation of sandwich panels with cellular core. *Composites Part B : Engineering*, 91 :458–472, 2016. (Cit  pages 5, 43, 48, et 86)
- [129] M. Montemurro, A. Pagani, G. A. Fiordilino, J. Pailh s, and E. Carrera. A general multi-scale two-level optimisation strategy for designing composite stiffened panels. *Composite Structures*, 201 :968–979, 2018. (Cit  pages 74, 75, 76, 77, 78, 92, 93, 103, 123, 142, 143, et 144)
- [130] M. D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2) :161–174, 1991. (Cit  pages 24, 25, 26, et 27)
- [131] J. M. Murphy, D. M. Sexton, D. H. Barnett, G. S. Jones, M. J. Webb, M. Collins, and D. A. Stainforth. Quantification of modelling uncertainties in a large ensemble of climate change simulations. *Nature*, 430(7001) :768–772, 2004. (Cit  page 28)
- [132] A. Nasir, N. Sims, and D. Wagg. Direct Normal Form Analysis of Oscillators with Different Combinations of Geometric Nonlinear Stiffness Terms. *Journal of Applied and Computational Mechanics*, 7(June) :1167–1182, 2021. (Cit  pages 123, 127, et 128)
- [133] C. Nguyen, X. Zhuang, L. Chamoin, X. Zhao, H. Nguyen-Xuan, and T. Rabczuk. Three-dimensional topology optimization of auxetic metamaterial using isogeometric analysis and model order reduction. *Computer Methods in Applied Mechanics and Engineering*, 371 :1–23, 2020. (Cit  page 43)
- [134] A. B. Owen. Randomly permuted (t,m,s)-nets and (t, s)-sequences. In H. Niederreiter and P. J.-S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 299–317, New York, NY, 1995. Springer New York. (Cit  page 182)
- [135] A. B. Owen. Sobol’ indices and shapley value. *SIAM-ASA Journal on Uncertainty Quantification*, 2(1), 2014. (Cit  page 58)
- [136] C. C. Pantelides. The Consistent Initialization of Differential-Algebraic Systems. *SIAM Journal on Scientific and Statistical Computing*, 9(2) :213–231, 1988. (Cit  page 189)
- [137] A. Pathak, K. Schneider, and V. Norrefeldt. Use of Modelica to predict risk of Covid-19 infection in indoor environments. *Proceedings of 14th Modelica Conference 2021, Link ping, Sweden, September 20-24, 2021*, 181 :463–469, 2021. (Cit  page 19)
- [138] L. Piegl and W. Tiller. *The NURBS books*. Springer-Verlag, New York, NY, USA, second edition, 1997. (Cit  pages 4, 38, 43, 85, 86, 87, 88, 90, 91, et 107)

-
- [139] L. A. Piegl and W. Tiller. Computing the derivative of NURBS with respect to a knot. *Computer Aided Geometric Design*, 15(9) :925–934, 1998. (Cité pages 104 et 107)
- [140] E. Plischke. An effective algorithm for computing global sensitivity indices (EASI). *Reliability Engineering & System Safety*, 95(4) :354–360, 2010. (Cité pages 29 et 57)
- [141] K. Plogert. The tailoring process in the German V-Model. *Journal of Systems Architecture*, 42 :601–609, 1996. (Cité page 16)
- [142] M. Prado-Velasco. In-silico virtual prototyping multilevel modeling system for Cyborgs (CybSim) as a novel approach for current challenges in biosciencies. *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*, 181 :485–496, 2021. (Cité page 19)
- [143] E. Pruliere, F. Chinesta, and A. Ammar. On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition. *Mathematics and Computers in Simulation*, 81(4) :791–810, dec 2010. (Cité page 36)
- [144] S. Razavi and H. V. Gupta. A new framework for comprehensive, robust, and efficient global sensitivity analysis : 1. Theory. *Water Resources Research*, 52(1) :423–439, 2016. (Cité page 24)
- [145] S. Razavi and H. V. Gupta. A new framework for comprehensive, robust, and efficient global sensitivity analysis : 2. Application. *Water Resources Research*, 52(1) :440–455, 2016. (Cité page 24)
- [146] S. Razavi, A. Jakeman, A. Saltelli, C. Prieur, B. Iooss, E. Borgonovo, E. Plischke, S. Lo Piano, T. Iwanaga, W. Becker, S. Tarantola, J. H. Guillaume, J. Jakeman, H. Gupta, N. Melillo, G. Rabitti, V. Chabridon, Q. Duan, X. Sun, S. Smith, R. Sheikholeslami, N. Hosseini, M. Asadzadeh, A. Puy, S. Kucherenko, and H. R. Maier. The Future of Sensitivity Analysis : An essential discipline for systems modeling and policy support. *Environmental Modelling and Software*, 137, 2021. (Cité pages 24 et 25)
- [147] C. Ren, Y. Aoues, D. Lemosse, and E. Souza De Cursi. Ensemble of surrogates combining Kriging and Artificial Neural Networks for reliability analysis with local goodness measurement. *Structural Safety*, 96(October 2021) :102186, 2022. (Cité pages 123, 127, et 128)
- [148] R. Renier and R. Chenouard. De SysML a Modelica : aide à la formalisation de modèles de simulation en conception préliminaire. In *12ème Colloque National AIP PRIMECA*, pages 1–9, Le Mont Dore, 2011. (Cité page 16)
- [149] T. Rodriguez, M. Montemurro, P. Le Texier, and J. Pailhès. Structural displacement requirement in a topology optimization algorithm based on isogeometric entities. *Journal of Optimization Theory and Applications*, 184 :250 – 276, 2020. (Cité pages 5, 43, 48, et 86)

- [150] R. Salomon. Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. A survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, 39(3) :263–278, 1996. (Cit  pages 110, 115, 123, 132, et 134)
- [151] A. Saltelli. Making best use of model evaluation to compute sensitivity indices. *Computer Physics Communications*, 145(2) :280–297, 2002. (Cit  pages 29, 53, 56, 57, 59, et 60)
- [152] A. Saltelli, P. Annoni, I. Azzini, F. Campolongo, M. Ratto, and S. Tarantola. Variance based sensitivity analysis of model output. design and estimator for the total sensitivity index. *Computer Physics Communications*, 181(2) :259–270, 2010. (Cit  pages 29, 53, 56, 57, 59, 60, et 61)
- [153] A. Saltelli and R. Bolado. An alternative way to compute Fourier amplitude sensitivity test (FAST). *Computational Statistics and Data Analysis*, 26(4) :445–460, feb 1998. (Cit  pages 57, 59, et 61)
- [154] A. Saltelli et al. *Global Sensitivity Analysis : The Primer*. Number 3. 2008. (Cit  pages 24 et 25)
- [155] A. Saltelli, S. Tarantola, and F. Campolongo. Sensitivity analysis as an ingredient of modeling. *Statistical Science*, 15(4) :377–395, 2000. (Cit  page 29)
- [156] G. Sarazin, J. Morio, A. Lagnoux, M. Balesdent, and L. Brevault. Reliability-oriented sensitivity analysis in presence of data-driven epistemic uncertainty. *Reliability Engineering & System Safety*, 215 :107733, 2021. (Cit  page 26)
- [157] K. J. Schlager. Systems Engineering—Key to Modern Development. *IRE Transactions on Engineering Management*, EM-3(3) :64–66, 1956. (Cit  page 14)
- [158] T. Schranz, C. M ldrup Legaard, D. Tola, and G. Schweiger. Portable runtime environments for Python-based FMUs : Adding Docker support to UniFMU. *Proceedings of 14th Modelica Conference 2021, Link ping, Sweden, September 20-24, 2021*, 181 :419–424, 2021. (Cit  page 23)
- [159] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games (AM-28), Volume II*, pages 3–17, 1953. (Cit  pages 29 et 58)
- [160] P. F. Shustin, S. Ubaru, V. Kalantzis, L. Horesh, and H. Avron. Pcenet : High dimensional surrogate modeling for learning uncertainty. 2022. (Cit  page 38)
- [161] J. Sieber and K. Bernd. Control based bifurcation analysis for experiments. *Nonlinear Dynamics*, 51 :365–377, 2008. (Cit  pages 68 et 71)
- [162] F. D. SILVA. *M thodologie de r duction de mod les multiphysiques pour la conception et la commande d’une chapine de traction  lectrique*. PhD thesis, Universit  Paris-Saclay, 2016. (Cit  page 34)
- [163] I. M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *U.S.S.R. Comput. Math. Math. Phys.*, 7 :86–112, 1967. (Cit  page 27)

-
- [164] I. M. Sobol. Sensitivity Estimates for Nonlinear Mathematical Models. *MMCE*, 1(4) :407–414, 1993. (Cité pages 24 et 25)
- [165] I. M. Sobol and Y. L. Levitan. On the use of variance reducing multipliers in Monte Carlo computations of a global sensitivity index. *Computer Physics Communications*, 117 :52–61, 1999. (Cité page 59)
- [166] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27(1) :1–31, 1978. (Cité page 113)
- [167] T. Sommer. Modelica-external library. <https://github.com/CATIA-Systems/Modelica-ExternalLibrary>, 2021. (Cité pages 48 et 180)
- [168] K. Sopian, A. H. Al-Waeli, and H. A. Kazem. Energy, exergy and efficiency of four photovoltaic thermal collectors with different energy storage material. *Journal of Energy Storage*, 29, 2020. Cited by : 56. (Cité page 158)
- [169] A. SOUPART-CARON. *Stockage de chaleur dans les Matériaux à Changement de Phase*. PhD thesis, I-MEP2, 2015. (Cité pages 158 et 160)
- [170] M. Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2) :143–151, 1987. (Cité page 182)
- [171] S. M. Stigler. The Epic Story of Maximum Likelihood. *A CONTROLLER*, 22(4) :598–620, nov 2007. (Cité page 39)
- [172] C. J. Stone. The use of polynomial splines and their tensor products in multivariate function estimation. *Annals of Statistics*, 6(4) :701–726, 1994. (Cité page 26)
- [173] Y. Sun, S. Vogel, and H. Steuer. Combining advantages of specialized simulation tools and modelica models using functional mock-up interface (fmi). In *Proceedings from the 8th International Modelica Conference, Technical University, Dresden, Germany*, pages 491–494, 2011. (Cité page 23)
- [174] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal of Optimization*, 12 :555–573, 2002. (Cité pages 93, 101, et 103)
- [175] S. Tarantola, D. Gatelli, and T. A. Mara. Random balance designs for the estimation of first order global sensitivity indices. *Reliability Engineering & System Safety*, 91(6) :717–727, 2006. (Cité pages 29 et 57)
- [176] R. Tarjan. Tarjan.pdf. *SIAM Journal on Scientific Computing*, 1(2) :146–160, 1972. (Cité page 189)
- [177] J. Y. Tissot and C. Prieur. Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliability Engineering & System Safety*, 107 :205–213, 2012. (Cité pages 29, 57, et 73)

- [178] C. J. Turner. *HyPerModels : hyperdimensional performance models for engineering design*. PhD thesis, University of Texas at Austin, 2005. (Cit  pages 46, 85, 91, et 108)
- [179] B. Vuillod, L. Hallo, E. Panettieri, and M. Montemurro. Sensitivity Analysis of a Car Shock Absorber Through a Functional Mock-up Units-Based Modelling Strategy. In *Proceedings of 14th Modelica Conference 2021, Link ping, Sweden, September 20-24, 2021*, pages 307–314, sep 2021. (Cit  pages 19 et 29)
- [180] B. Vuillod, M. Montemurro, E. Panettieri, and L. Hallo. A comparison between Sobol’s indices and Shapley’s effect for global sensitivity analysis of systems with independent input variables. *Reliability Engineering and System Safety*, 234(February) :109177, 2023. (Cit  pages 29, 47, 52, 123, et 162)
- [181] B. Vuillod, E. Panettieri, L. Hallo, and M. Montemurro. Preliminary design of non-linear systems based on global sensitivity analysis and modelica language. In *Proceeding of the 24th International Conference on Engineering Design*, pages 2925–2934, 2023. (Cit  page 29)
- [182] H. Wackernagel. *Multivariate Geostatistics (Google eBook)*. page 387, 2003. (Cit  page 41)
- [183] G. Wahba. Spline Models for Observational Data. In *CBMS-NSF Regional Conference Series in Applied Mathematics*, Philadelphia, PA, 1990. Society for Industrial and Applied Mathematics. (Cit  page 94)
- [184] Q. Wang, L. Yang, and J. Song. Preparation, thermal conductivity, and applications of nano-enhanced phase change materials (NEPCMs) in solar heat collection : A review. *Journal of Energy Storage*, 63(March) :107047, 2023. (Cit  pages 158 et 160)
- [185] I. Waterloo Maple. *MapleSim User’s Guide*. Maplesoft, 2022. (Cit  page 64)
- [186] J. E. Whitehouse. *Circuit analysis*. 1997. (Cit  page 19)
- [187] M. Wiens, T. Meyer, and P. Thomas. The Potential of FMI for the Development of Digital Twins for Large Modular Multi-Domain Systems. volume 181, pages 235–240, 2021. (Cit  page 23)
- [188] Z. Wu, D. Wang, W. Wang, K. Zhao, H. Zhou, and W. Zhang. Hybrid meta-model of radial basis function and polynomial chaos expansions with orthogonal constraints for global sensitivity analysis. *Structural and Multidisciplinary Optimization*, 62 :597–617, 2020. (Cit  pages 38 et 39)
- [189] Z. Wu, W. Wang, D. Wang, K. Zhao, and W. Zhang. Global sensitivity analysis using orthogonal augmented radial basis function. *Reliability Engineering & System Safety*, 185 :291–302, 2019. (Cit  pages 58, 59, 60, 61, et 72)
- [190] A. W. Wymore. *Model-based systems engineering*. 1993. (Cit  page 15)

- [191] Y. Xinxin, S. Jian, G. Weijie, L. Min, and D. Libin. An adaptative pce-hdmr meta-modeling approach for high-dimensional problems. *Structural and Multidisciplinary Optimisation*, 2021. (Cité page 38)
- [192] J. Yang and P. Li. Brain networks of explicit and implicit learning. *PLOS ONE*, 7(8) :1–9, 08 2012. (Cité page 43)
- [193] B. Yegnanarayana. *Artificial Neural Networks*. Prentice-Hall of India Pvt.Ltd, 2004. (Cité page 43)
- [194] J. Yu, Y. Yang, X. Yang, Q. Kong, Y. Liu, and J. Yan. Effect of porous media on the heat transfer enhancement for a thermal energy storage unit. *Energy Procedia*, 152 :984–989, 2018. (Cité page 158)
- [195] C. Zenger. Sparse Grid, 1991. (Cité page 181)
- [196] K. Zhang, Z. Lu, K. Cheng, L. Wang, and Y. Guo. Global sensitivity analysis for multivariate output model and dynamic models. *Reliability Engineering & System Safety*, 204 :107195, 2020. (Cité page 26)
- [197] S. Zhao, N. Zhang, X. Zhou, and L. Zhang. Particle shape effects on fabric of granular random packing. *Powder Technology*, 310 :175–186, 2017. (Cité page 18)
- [198] C. Zhou, Z. Shi, S. Kucherenko, and H. Zhao. A unified approach for global sensitivity analysis based on active subspace and Kriging. *Reliability Engineering & System Safety*, 217 :108080, 2022. (Cité page 29)

Stratégie de modélisation multi-fidélité via une approche système incluant des métamodèles basés sur les entités NURBS

Résumé

Plus un problème est complexe, plus la quantité de ressources informatiques nécessaires pour le simuler est importante. D'autre part, le besoin de précision dans les résultats d'un système ne sera pas le même selon sa phase de conception et le domaine étudié. L'objectif de cette thèse est de proposer une stratégie de modélisation multi-fidélité rapide et peu coûteuse en ressources de calcul. Pour répondre à ce besoin, une modélisation hybride est développée, couplant l'approche Model-Based System Engineering (MBSE) et le métamodèle basé sur les hypersurfaces Non-Uniform Rational Basis-Spline (NURBS). Plus précisément, l'enjeu scientifique de ce travail est le développement du métamodèle basé sur les entités NURBS pour simuler le comportement de systèmes fortement non-linéaires nécessitant une modélisation haute-fidélité mais capable de fournir les résultats en temps réel pour être compatible avec l'approche MBSE. Dans ce contexte, le métamodèle basé sur les entités NURBS est obtenu comme solution d'un problème d'optimisation résolu avec un algorithme au gradient. En outre, un terme de lissage est intégré dans la formulation du problème pour non seulement réduire l'influence d'éventuelles non-linéarités parasites de la base de données d'entraînement mais également pour limiter le phénomène d'overfitting. L'enjeu technicoscientifique de ce travail est de parvenir à coupler l'approche générale MBSE avec le métamodèle à base de NURBS.

Mots clefs :

Métamodèle ; NURBS ; Approche MBSE ; Multi-fidélité ; Multi-Physique ; Analyse de Sensibilité Globale

Résumé en anglais

The more complex the problem, the greater the amount of computational resources needed to simulate it. On the other hand, the need for accuracy in the results of a system will not be the same depending on its design phase and the domain studied. The goal of this thesis is to propose a fast, low-cost multi-fidelity modeling strategy. To meet this need, a hybrid modeling approach is developed that combines Model-Based System Engineering (MBSE) and a metamodel based on Non-Uniform Rational Basis-Spline (NURBS) hypersurfaces. More specifically, the scientific challenge of this work is to develop a metamodel based on NURBS entities to simulate the behavior of highly nonlinear systems that require high fidelity modeling but are capable of providing results in real time to be compatible with the MBSE approach. In this context, the NURBS entity-based metamodel is obtained as a solution to an optimization problem solved with a gradient algorithm. In addition, a smoothing term is included in the problem formulation, not only to reduce the influence of any spurious nonlinearities in the training database, but also to limit the phenomenon of overfitting. The technical and scientific challenge of this work is to couple the general MBSE approach with the NURBS-based metamodel.

Keywords:

Metamodel ; NURBS ; MBSE approach ; Multi-fidelity ; Multi-Physic ; Global Sensitivity Analysis