

Numéro d'ordre : 2014-13

THÈSE DE L'UNIVERSITÉ DE LYON

présentée pour obtenir le grade de

DOCTEUR DE L'ÉCOLE CENTRALE DE LYON

en mécanique

par

Pierre FROMENT

Optimisation de formes paramétriques en grande dimension

Présentée et soutenue publiquement le 24 avril 2014 devant le jury composé de :

Marc ALBERTELLI, Renault S.A., Ingénieur de recherche

Alain COMBESCURE, LaMCoS - INSA Lyon, Professeur des universités

Frédéric GILLOT, École Centrale de Lyon, Maître de conférences

Louis JÉZÉQUEL, École Centrale de Lyon, Professeur des universités

Roger OHAYON, LMSSC - CNAM, Professeur émérite

Christian SOIZE, Laboratoire MSME - Université de Paris-Est, Professeur des universités

Examineur

Président du jury

Co-encadrant

Directeur de thèse

Rapporteur

Rapporteur



ÉCOLE CENTRALE DE LYON
LTDS - D2S

36, avenue Guy de Collongue
69130 Écully



TECHNOCENTRE RENAULT
Direction de la recherche

1, avenue du golf
78280 Guyancourt

Ce manuscrit est confidentiel pour une durée de 3 ans.

Résumé

Les méthodes actuelles d'optimisation de formes permettent des gains importants vis-à-vis des fonctions à optimiser. Elles sont largement utilisées par les industriels, notamment dans l'automobile comme chez Renault. Parmi ces approches, l'optimisation de formes paramétriques permet d'obtenir rapidement une géométrie optimale sous réserve que l'espace de conception soit assez restreint pour pouvoir être parcouru en un temps de simulation raisonnable. Ce manuscrit présente deux méthodes d'optimisation de formes paramétriques en grande dimension pour des applications nécessitant des coûts de calculs importants, par exemple en mécanique des fluides.

Une manière originale de reconstruire un modèle CAO paramétré à partir d'une surface morte est présentée au préalable. Nous proposons une approche pour identifier les zones interdites de l'espace de conception ainsi que leurs gestions dans une boucle d'optimisation par plans d'expériences. La première méthode s'appuie sur des techniques statistiques pour lever le verrou du nombre de degrés de liberté et utilise une optimisation à deux niveaux de fidélité pour minimiser les temps de calcul. Cette méthode en rupture avec le processus industriel habituel a été appliquée pour optimiser le coefficient de traînée aérodynamique d'un véhicule.

La seconde méthode se base sur l'exploitation des gradients fournis par les solveurs adjoints, c'est-à-dire sur les sensibilités du critère (comme l'uniformité d'un écoulement par exemple) par rapport aux degrés de liberté de l'optimisation. Cette méthode innovante et en rupture avec les approches classiques permet de lever très naturellement le verrou du nombre de paramètres. Cependant, les gradients fournis par les logiciels ne sont pas donnés par rapport aux paramètres CAO mais par rapport aux nœuds du maillage. Nous proposons donc une façon d'étendre ces gradients jusqu'aux paramètres CAO. Des exemples académiques ont permis de montrer la pertinence et la validité de notre approche.

Mots clés : optimisation de formes, CAO, paramétrisation, analyse de sensibilité

Abstract

Current design loops for shape optimizations allow significant improvements in relation to the functions that need to be optimized. They are widely used in industry, particularly in the car industry like Renault. Among these approaches, parametric shape optimization allows rapid enhancement of the shape, on the condition that the design space is confined enough in order to be explored within a reasonable computational time. This Thesis introduces two **CAD**-based large-scale shape optimization methods for products requiring significant computational cost, for example in fluid mechanics.

An original way to create a parameterized **CAD** model developed from a dead geometry is presented first. We propose an approach to identify the restricted areas of the design space and their managements in an optimization loop that uses a design of experiments. The first method is based on statistical techniques to circumvent the difficulties of large-scale optimizations and uses a two-level multi-fidelity modelling approach to minimize the computational time. This method, breaking away from the usual industrial process, was applied to optimize the aerodynamic drag coefficient of a car body.

The second method is based on the gradients provided by adjoint solvers, that is to say on the sensitivity of the cost function (such as the uniformity deviation for example) with respect to the design points or displacement boundaries. This innovative method breaking away from classical approaches naturally gets over the number of degrees of freedom. However, the sensitivities provided by softwares are not computed with respect to **CAD** parameters but with respect to the coordinates of the vertices of the surface mesh. Thus, we propose a way to extend these gradients to **CAD** parameters. Academic test cases have proved the efficiency and accuracy of our method.

Keywords: shape optimization, **CAD**, parameterization, sensitivity analysis

*« Ce que l'on conçoit bien s'énonce clairement,
Et les mots pour le dire arrivent aisément. »*

Nicolas BOILEAU

Remerciements

Trois ans... De ces années qui resteront à jamais gravées dans la mémoire de Pierre Froment tant elles représentent une somme impressionnante d'efforts et d'épreuves de toutes sortes sans jamais lâcher le morceau, sans jamais douter. Et puis, il y a eu aussi ces moments inoubliables d'espoirs ponctués de rebondissements et de rires toujours partagés sur le terrain avec les membres de son club. L'aventure humaine est tellement passionnante.

Vous l'avez compris, aujourd'hui, Pierre Froment a répondu présent en finale. Le jeune joueur n'a pas douté, il savait qu'il faisait partie d'un grand club, d'un club ambitieux. Et pourtant, tout n'a pas été facile...

Après une enfance passée sur les terrains du pays de Caux, Pierre signe son premier contrat professionnel en classes préparatoires à Rouen. En septembre 2006, il décide de rejoindre le club de Toulon pour se frotter aux réalités du terrain. Deux ans plus tard, il tente l'aventure à l'étranger pour parfaire sa formation : son choix se portera sur la rude mais efficace recherche allemande. De retour en France en 2010, le jeune joueur est libre de tout engagement. Il a beaucoup progressé, mais il lui manque l'essentiel : le soutien d'un grand club. Jusqu'à ce jour d'avril 2010 où il reçoit un courriel de Maryan Sidorkiewicz. L'aventure peut commencer. Confidences...

C'est d'abord la rencontre avec Louis Jézéquel, le président du club. Un homme de contact exceptionnel – bien secondé par Isabelle Tixier, sa mémoire vivante – qui n'a pas son pareil pour tirer le meilleur de ses poulains. Et puis, « jamais le dernier pour se faire une bonne bouffe »... Leur première rencontre s'est faite dans le mythique stade du Technocentre en présence du directeur général délégué, Frédéric Gillot. Pierre en dira le plus grand bien : « Il m'a énormément apporté, en me montrant le chemin à suivre, par ses précieux conseils techniques et... son humour décapant ! »

Il faut dire que le recrutement est à la hauteur des objectifs : Vaumoron, Mercier, Maillot, bientôt renforcés par Albertelli. « Une montée en puissance qui ne laisse planer aucun doute sur le niveau des objectifs du club » déclare alors Pierre, avant d'ajouter que « Marc apporte énormément à l'équipe, par ses qualités techniques de niveau international ». Un milieu de terrain hors-pair, dont la générosité dans le jeu se retrouve au jour le jour. Un autre artisan de la montée en puissance du groupe a été Vaumoron. Le sarthois a offert un récital à l'entre-jeu. Il était partout. Et très facile. Les cadres veillent au grain, à l'image de Mercier, « un personnage jovial et attachant », ou encore de Maillot dont « la générosité débordante n'a d'égal que la gentillesse ». Le tout sous le regard bienveillant du vétéran de l'équipe, Fourcade : les tempes un peu plus clairsemées, les jambes un peu moins vaillantes, mais toujours la recherche au cœur. Sans oublier l'opiniâtre expert technique Tourbier « dont le caractère entier n'est plus à démontrer. Il a su me faire totalement confiance » confie le dieppois.

Surtout, l'équipe peut s'appuyer sur sa jeune garde. Cilia et Leblond – issus du centre de formation – ont marqué les esprits dès les premiers instants. Ils se mettent vite au diapason. « Ils ont permis à l'équipe de prendre une autre dimension ! C'est tellement facile de jouer avec eux. »

Tout a été fait pour mettre Pierre dans les meilleures conditions afin d'être en mesure de se confronter au gratin mondial : une première conférence à São Paulo en juillet 2012. « Des heures de labeur à Guyancourt, une mise au vert aux couleurs brésiliennes, on ne pouvait devenir que complices avec Paul. » La deuxième conférence – en mai 2013 – a emmené Dapogny, Michailidis et Froment à Orlando. « On est d'abord allés se ressourcer en Californie afin de mettre tous les atouts de notre côté pour affronter les T-rex, Hulk, Harry Potter... et mêmes des dauphins » ajoute avec une pointe d'ironie Pierre. Une virée américaine inoubliable.

Quant aux supporters, Pierre ne tarit pas d'éloges pour ceux qui se muent en 12^e homme de l'équipe, « avec notamment une pensée particulière envers le groupe des Ultras – mes intimes – et aussi envers le kop de Guyancourt qui m'a soutenu au quotidien. »

Il n'y a pas de grand match sans un arbitrage à la hauteur... Ce fut le cas du trio des hommes en noir formé par messieurs Combescure, Ohayon et Soize dont les interventions et la bienveillance furent les garants d'une finale réussie.

De même, il est des gens que l'on voit peu dans les stades ni même autour, mais qui apportent dans l'ombre leur contribution à l'édifice. « Un grand merci à ma famille! Avec une pensée particulière à ma tante Sylvie pour les ajustages techniques. Et bien évidemment à mes frères et sœurs Antoine et Claire pour leur présence... Il aura fallu tant d'investissements de la part de mes parents pour que je touche le haut niveau. De mes premières passes à la préparation de la finale, que d'épopées... »

Et le normand de terminer : « que les gens cités dans cet article en soient vivement remerciés... » Vive le sport !

Table des matières

Remerciements	xiii
Introduction	1
1 Introduction à l'optimisation de formes surfaciques	3
1.1 Ingénierie des systèmes et cycle en V	3
1.1.1 Architecture fonctionnelle	6
1.1.2 Architecture organique	7
1.1.3 Avantages et limites de la méthode	7
1.2 Familles de paramétrage de forme	8
1.2.1 Paramètres métier pilotés par CAO	9
1.2.2 Paramètres métier pilotés par morphing	11
1.2.3 Paramétrage dédié à l'optimisation isogéométrique	11
1.2.4 Paramétrage dédié à l'optimisation topologique	12
1.2.5 Paramétrage dédié à l'optimisation géométrique	13
1.2.6 Variante d'optimisation topologique dédiée à la CFD	13
1.3 Méthodes d'optimisation (<i>gradient-based</i> ou <i>gradient-free</i>)	16
1.3.1 Méthodes d'optimisation par plans d'expériences et surfaces de réponse (<i>gradient-free</i>)	18
1.3.2 Méthode des dérivées d'ordre supérieur	24
1.3.3 Optimisation directe (<i>gradient-based</i>)	26
1.4 Synthèse	30
Bibliographie du 1 ^{er} chapitre	32
2 Réduction de la dimension du problème et multiniveaux de fidélité	35
2.1 Construction d'un modèle paramétré	36
2.1.1 Contexte	36
2.1.2 Reconstruction surfacique	38
2.1.3 Paramétrisation	44
2.1.4 Application à un cas industriel	46

2.2	Analyse hors simulation d'un modèle paramétré	46
2.2.1	Détection des zones interdites	46
2.2.2	Gestion des zones interdites par machine à vecteurs de support	51
2.2.3	Gestion des zones interdites par génération d'un grand nombre d'essais	57
2.3	Réduction du nombre de paramètres <i>a priori</i>	58
2.3.1	1 ^{re} voie : voxellisation et analyse en composantes principales	58
2.3.2	2 ^e voie : réseau de neurones artificiels	67
2.4	Approche multi-niveaux de fidélité	69
2.5	Application de la méthode à un cas industriel	70
2.5.1	Description du cas industriel	71
2.5.2	Construction du modèle CAO	72
2.5.3	Modélisation de la frontière du domaine de conception	74
2.5.4	Création des méta-paramètres	75
2.5.5	Génération du plan d'expériences	76
2.5.6	Simulations	77
2.5.7	Gestion de la boucle de calcul	78
2.5.8	Résultats	79
2.6	Discussion de l'application	80
	Bibliographie du 2 ^e chapitre	83
3	Calcul de la sensibilité d'une prestation par rapport aux paramètres métier	85
3.1	État de l'art	86
3.2	Méthode proposée : morphing d'une forme à isoconnectivité	89
3.2.1	Paramétrisation d'une surface triangulée	90
3.2.2	Choix de la méthode	92
3.2.3	Transformations harmoniques	93
3.3	Calcul des sensibilités en utilisant des transformations harmoniques	95
3.3.1	Description théorique	95
3.3.2	Procédure de calcul de la sensibilité	96
3.3.3	Application sur un cas académique	97
3.4	Mise en place de la démarche	97
3.4.1	Présentation du cas d'étude	97
3.4.2	Choix du domaine paramétrique	98
3.4.3	Point invariant en 2D et sens de parcours de la frontière	100
3.4.4	Calcul du gradient au niveau des nœuds ou des éléments	101

3.4.5	Valeurs des perturbations	103
3.5	Application de la méthode	104
3.5.1	Boucle d'optimisation	104
3.5.2	Cas d'étude	106
3.6	Discussion des résultats	115
	Bibliographie du 3 ^e chapitre	118
Conclusion et perspectives		121
A Quelques définitions		123
A.1	Continuité géométrique	123
B Plans d'expériences		125
B.1	Plans d'expériences élémentaires	125
B.1.1	Plans de Monte-Carlo	125
B.1.2	Plans hypercubes latins	126
B.2	Plans d'expériences avancés	129
B.2.1	Substitution des points non faisables	129
B.2.2	Plans D-optimaux	129
B.2.3	Plans G-optimaux	130
B.2.4	Plans d'expériences auto-adaptatifs	130
B.3	Evaluation des performances d'un plan	135
B.3.1	Nombre de coefficients à estimer	135
B.3.2	Estimation des coefficients pour des modèles polynomiaux	135
B.3.3	Variance de prédiction	136
B.3.4	Critères de comparaison	137
	Bibliographie des annexes	139
Bibliographie globale		146
Liste des acronymes et abréviations		149

Table des figures

1.1	Description du cycle en V	4
1.2	Prédictivité des modèles en fonction de l'avancement dans le cycle en V	5
1.3	Étapes du déploiement du cahier des charges lors de la descente du cycle en V	6
1.4	Exemple d'une CAO	9
1.5	Exemple d'une boucle d'optimisation paramétrique	10
1.6	Illustration d'un morphing	11
1.7	Enrichissement d'une géométrie par raffinement h-p-k	12
1.8	Illustration d'une optimisation topologique	13
1.9	Illustration d'une optimisation géométrique sur une console	14
1.10	Zones de refoulement	14
1.11	Sédimentation des cellules et tolérance de refoulement	15
1.12	Influence du niveau de coupure	16
1.13	Exemple de mise en données d'un problème d'optimisation de forme	17
1.14	Schéma représentant une méthode d'optimisation par plan d'expériences	19
1.15	Fonctions charnières des modèles MARS	21
1.16	Exemple de prédiction par krigeage en dimension 1	23
1.17	Illustration du front de Pareto pour un problème bi-objectif	24
1.18	Influence de chaque dérivée	28
1.19	Schéma des deux approches pour parvenir aux équations adjointes discrètes	29
2.1	Paramètres d'équilibre du sillage d'un véhicule	36
2.2	Sélection du pavillon	38
2.3	Extraction de la surface à reconstruire	39
2.4	Transport-projection sur le carré unité	39
2.5	Recherche des fonctions d'interpolation des méthodes d'éléments finis	40
2.6	Discrétisation de la surface de base	42
2.7	Création du réseau de courbes	43
2.8	Création de la surface sur réseau	43
2.9	Déformation de la courbe du pavillon en $y = 0$	44

2.10	Schéma de la reconstruction surfacique	45
2.11	Reconstruction surfacique de Clio iv	46
2.12	Évolution de la flèche du pavillon	47
2.13	Évolution du porte-à-faux arrière	48
2.14	Plaque percée	49
2.15	Analyse des résultats avec 2 paramètres	50
2.16	Analyse des résultats avec 3 paramètres	51
2.17	Illustration en dimension 2 de l'hyperplan séparateur	52
2.18	Influence de la constante de pénalisation	56
2.19	Influence de l'ordre du polynôme	57
2.20	Influence de σ	58
2.21	Pixellisation d'une forme quelconque	59
2.22	Plan généré en dimension 3	61
2.23	Estimation de la dimension du problème par le maximum de vraisemblance	62
2.24	Représentation des coordonnées des points sur les trois premiers axes	62
2.25	Tube	63
2.26	Tube voxellisé (grossièrement)	64
2.27	Représentation des trois premières composantes principales	65
2.28	Représentation des trois premières composantes principales autour d'une position	66
2.29	Détail des sous-réseaux	67
2.30	Réseau de neurones artificiels auto-associatif	68
2.31	Version symbolique du 1 ^{er} sous-réseau	69
2.32	Schéma-bloc de la méthode	71
2.33	Reconstruction surfacique d'un modèle « archi »	73
2.34	Répartition des essais pour le triplet de paramètres	75
2.35	Étude d'un paramètre particulier	76
2.36	Positionnement des zones de raffinement à l'arrière du véhicule	78
2.37	Visualisation du coefficient de pression adimensionné	80
2.38	Visualisation des lignes de courant (vitesse)	81
3.1	Exemple de visualisation des sensibilités en mécanique des fluides	86
3.2	Notations	87
3.3	Processus d'évaluation de la sensibilité d'une forme vis-à-vis du paramètre α_2	88
3.4	Description schématique de la méthode proposée dans [Robinson 2012]	89
3.5	Illustration des problèmes de la projection selon la normale	90

3.6	Illustration du dépliage du masque de Toutânkhamon sur un disque	91
3.7	Exemple du genre de deux surfaces	91
3.8	Maillage des formes initiale et perturbée	95
3.9	Transformations harmoniques des maillages initial et perturbé	96
3.10	Morphing du maillage de la forme initiale sur le maillage de la forme perturbée	96
3.11	Construction de $\tilde{\Omega}(\alpha_i + \delta\alpha_i)$	96
3.12	Translation d'un cube et gradients associés	98
3.13	Coude avec un paramètre	98
3.14	Choix du domaine paramétrique	99
3.15	Comparaison des maillages du coude initial et du coude perturbé	99
3.16	Découpage de la géométrie en deux parties homéomorphes à un disque	99
3.17	Description schématique de l'évaluation des sensibilités	100
3.18	Rotation des disques paramétriques	100
3.19	Paramétrisation du demi-coude sur un disque unité	101
3.20	Visualisation des gradients	102
3.21	Schéma de la boucle d'optimisation directe	104
3.22	Encastrement de la base du coude	107
3.23	Paramétrage du coude avec 7 sections	108
3.24	Surface de découpe de la géométrie	109
3.25	Courbes des résultats du cas n° 1	109
3.26	Évolution de la forme du coude à sections circulaires au fil des itérations	110
3.27	Visualisation de la sensibilité du maillage par rapport à un paramètre CAO	110
3.28	Évolution de la forme du coude à sections rectangulaires au fil des itérations	111
3.29	Courbes des résultats du cas n° 5	112
3.30	Mise en données du problème en raideur statique : encastrement de la base et création du réseau d'éléments rigides.	112
3.31	Courbes des résultats du cas n° 7	113
3.32	Évolution de la forme du coude à sections circulaire au fil des itérations	113
3.33	Amélioration du paramétrage du coude	114
3.34	Paramétrage du coude avec 12 sections	115
3.35	Courbes des résultats du cas n° 10	116
3.36	Diversité des cas d'étude	116
3.37	Sensibilité du maillage par rapport à la flèche du pavillon	117
A.1	Continuité G^1	123

B.1	Plan de Monte-Carlo en dimension 2 avec 20 essais	126
B.2	Plans LHS en dimension 2 avec 20 essais	127
B.3	Complètement d'un plan LHS en 2D	131
B.4	Répartition des essais en 5D	131
B.5	Duplication des essais dans l'hypercube	132
B.6	Répartition des essais en 5D avec la nouvelle distance	134
B.7	Comparaison des différents types de plan d'expériences	138

Liste des tableaux

1.1	Comparaison entre les différents types de paramétrage	9
2.1	Valeurs propres et qualité globale de reconstruction du plan avec des relations linéaires entre les données	60
2.2	Valeurs propres et qualité globale de reconstruction du plan avec des relations non linéaires entre les données	61
2.3	Les 20 premières valeurs propres et la qualité globale de reconstruction du tube .	65
2.4	Valeurs du SC_x	80
3.1	Précisions des systèmes CAO en fonction de la taille des modèles	104
3.2	Maximisation de la fréquence du premier mode propre (cas n° 1 à 4)	108
3.3	Maximisation de la fréquence du premier mode propre (cas n° 5 et 6)	111
3.4	Minimisation du déplacement du nœud maître (cas n° 7 et 8)	113
3.5	Minimisation du déplacement du nœud maître (cas n° 9 à 13)	114
B.1	Evolution du nombre de coins et de bords avec la dimension	132
B.2	Nombre de coefficients à estimer	135

Introduction

La mission d'un bureau d'études est de concevoir au plus juste tout en respectant un cahier des charges. Pour ce faire, il peut compter sur son expérience mais il peut surtout s'appuyer sur des méthodes d'optimisation. Ces dernières sont apparues historiquement dans l'automobile pour des contraintes liées aux prestations de NVH (*noise, vibration, and harshness*) et de tenue mécanique. Ces domaines ont l'avantage de ne pas nécessiter de puissance de calcul trop importante, ou du moins de pouvoir être simplifiés pour obtenir des résultats rapidement.

D'autres domaines n'ont en revanche pas pu bénéficier de ce type d'approche. C'est le cas par exemple de la mécanique des fluides où les modèles de simulation nécessitent une puissance de calcul considérable pour bénéficier d'une précision satisfaisante. Les variables d'optimisation correspondent aux paramètres géométriques qui sont issus d'un système de *conception assistée par ordinateur* (CAO). A titre d'exemple, en aérodynamique externe, ce sont les caractéristiques *design* (rayon, longueur, angle, ratio...) qui sont en grand nombre. Ainsi, évaluer l'influence de chaque paramètre sur des critères utilisant des modèles de simulation complexes est inenvisageable d'un point de vue du coût de calcul.

C'est dans ce contexte que nous nous plaçons. Par conséquent, deux verrous doivent être levés pour envisager des optimisations :

- o la grande dimension de l'espace des paramètres ;
- o la gestion du coût de calcul.

Pour y répondre, nous avons développé deux méthodes qui sont présentées dans ce manuscrit. La première est une extension des méthodes classiques basées sur des plans d'expériences. La seconde, plus spécifique aux prestations issues de la mécanique des fluides, repose sur une optimisation directe exploitant un calcul de sensibilités dans ce domaine.

Le premier chapitre (p. 3) retrace l'état de l'art sur les différentes familles de paramétrage et sur les méthodes d'optimisation. La façon de concevoir un produit dans l'industrie y sera également décrite.

Le deuxième chapitre (p. 35) est dédié à la présentation d'une méthode originale de reconstruction d'un modèle CAO paramétré à partir d'une surface morte. Nous proposons une approche pour identifier les zones interdites de l'espace de conception ainsi que leurs gestions dans une boucle d'optimisation par plans d'expériences. Pour lever la contrainte du nombre de degrés de liberté, nous proposons une méthode pour réduire la dimension du problème en s'appuyant sur des techniques statistiques. Enfin, nous mettons en place une optimisation à deux niveaux de fidélité pour s'affranchir des problèmes liés au temps de calcul. La boucle d'optimisation proposée a été appliquée sur un exemple industriel.

La seconde méthode, détaillée dans le troisième chapitre (p. 85), s'appuie sur les gradients, c'est-à-dire les sensibilités du critère (par exemple, une perte de charge) par rapport aux degrés de liberté de l'optimisation. En mécanique des structures (plus précisément dans le cadre des

Introduction

analyses linéaires), on exploite déjà ces informations pour optimiser des problèmes de grande taille. Depuis peu, les éditeurs en mécanique des fluides essaient de fournir ces gradients. Il est dès lors indispensable de construire les algorithmes qui sauront les exploiter. Cependant, les gradients fournis ne sont pas donnés par rapport aux paramètres CAO mais par rapport aux nœuds du maillage. L'essentiel du travail réalisé a justement consisté à étendre ces gradients jusqu'aux paramètres CAO. Grâce aux gradients, la dimension du problème ne compte plus, ce qui justifie amplement que l'on explore cette voie avec le plus grand intérêt. Nous avons montré la pertinence et la validité de notre approche au travers d'exemples académiques.

Introduction à l'optimisation de formes surfaciques

Sommaire

1.1	Ingénierie des systèmes et cycle en V	3
1.1.1	Architecture fonctionnelle	6
1.1.2	Architecture organique	7
1.1.3	Avantages et limites de la méthode	7
1.2	Familles de paramétrage de forme	8
1.2.1	Paramètres métier pilotés par CAO	9
1.2.2	Paramètres métier pilotés par morphing	11
1.2.3	Paramétrage dédié à l'optimisation isogéométrique	11
1.2.4	Paramétrage dédié à l'optimisation topologique	12
1.2.5	Paramétrage dédié à l'optimisation géométrique	13
1.2.6	Variante d'optimisation topologique dédiée à la CFD	13
1.3	Méthodes d'optimisation (<i>gradient-based</i> ou <i>gradient-free</i>)	16
1.3.1	Méthodes d'optimisation par plans d'expériences et surfaces de réponse (<i>gradient-free</i>)	18
1.3.1.1	Plans d'expériences	18
1.3.1.2	Surfaces de réponse	20
1.3.1.3	Optimisation	22
1.3.2	Méthode des dérivées d'ordre supérieur	24
1.3.2.1	Principe	24
1.3.2.2	Avantages et inconvénients	25
1.3.3	Optimisation directe (<i>gradient-based</i>)	26
1.3.3.1	Principe	26
1.3.3.2	Méthode adjointe en mécanique des fluides	27
1.3.3.3	Calcul des gradients par différences finies	27
1.3.3.4	Calcul des gradients par l'équation adjointe	27
1.4	Synthèse	30
	Bibliographie du 1^{er} chapitre	32

1.1 Ingénierie des systèmes et cycle en V

Issue des domaines de l'aéronautique, du spatial et de la défense, l'ingénierie des systèmes est dorénavant diffusée dans la plupart des domaines économiques : le transport, l'énergie, le

médical... Elle a pour objectif de formaliser et d'appréhender la conception de systèmes complexes sur la base de compromis entre des objectifs contradictoires. Par *système complexe*, on entend un ensemble d'éléments de technologies variées (mécanique, électrique, électronique, informatique...) dépendants les uns des autres. Le moteur en faveur du déploiement et de la mise en œuvre de cette méthode est porté par la nécessité de ne pas repartir de zéro, plus communément appelé *carry-over* (effet de report) dans l'ingénierie, par la nécessité de trouver une solution adaptée répondant aux besoins exprimés, aux délais et aux coûts, par la nécessité de démanteler les systèmes et réduire les déchets ultimes [Luzeaux 2011].

L'ingénierie des systèmes prend classiquement, en milieu industriel, la forme d'un cycle en V [Haskins 2007]. Le concept de base est de procéder à un découpage hiérarchisé entre systèmes, sous-systèmes, organes et pièces. C'est une combinaison de *top-down* caractérisant la phase de conception et de développement et de *bottom-up* caractérisant les tests et la validation [Lebrun 2003] (cf. figure 1.1). Ce cycle en V couvre le cycle de vie du produit, depuis les spécifications des clients jusqu'à la production en série.

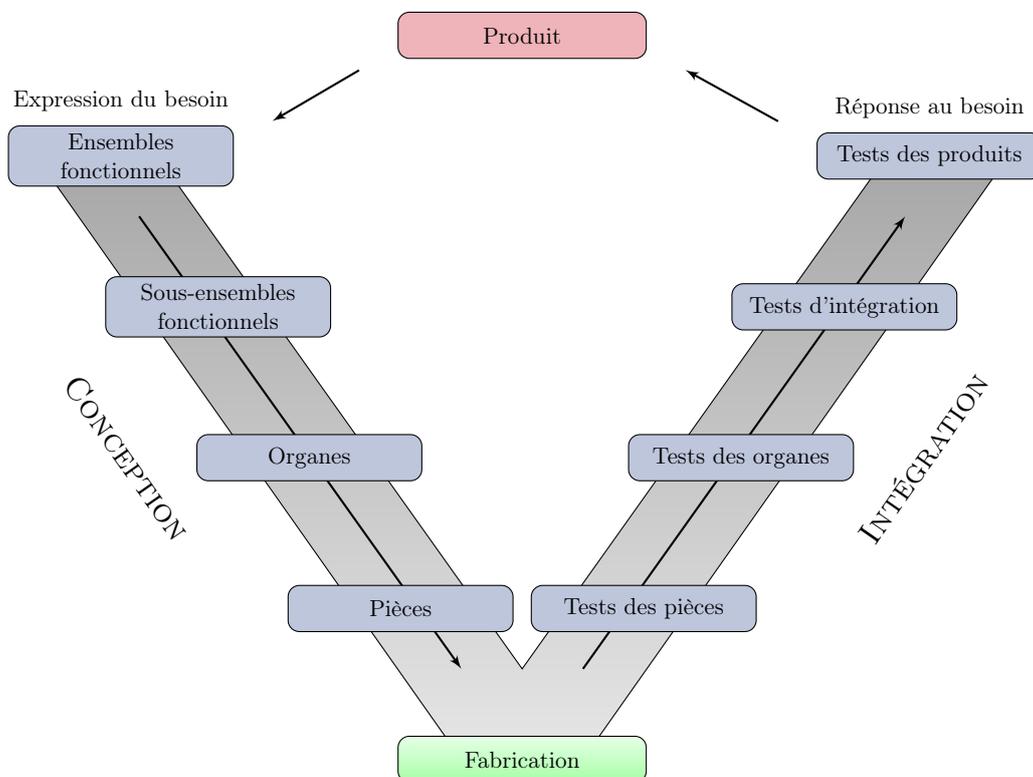


FIGURE 1.1 – Description du cycle en V

Le cycle en V organise la conception sur un double espace. D'une part, le système est envisagé d'un point de vue global vers une représentation locale. D'autre part, chaque fonction identifiée est affectée à un composant physique précis. La conception consiste en un aller-retour d'une vision globale et fonctionnelle à une représentation locale (détaillée) et physique (réalisable par les composants identifiés). La démarche de développement d'un système consiste dès lors à répartir les tâches d'ingénierie et les tâches d'intégration sur les deux branches du V :

1. les tâches de spécification se situent sur la branche descendante du V. Cette première phase de conception permet de spécifier et de concevoir chaque caractéristique du produit ;

2. les tâches de réalisation sont situées en bas du V. Elles consistent soit à fabriquer directement les constituants, soit à réutiliser des constituants ayant été développés précédemment, soit à acheter des constituants sur étagère ;
3. les tâches d'intégration sont sur la branche montante du V. Elles assemblent chacune des briques réalisées, et valident leur intégration.

À chaque étape de la conception, cette démarche permet de définir, pour chaque ensemble analysé, un domaine paramétrique, qui prend en compte, sous forme synthétique, le comportement des autres ensembles considérés comme extérieurs. Ainsi, à chaque étape, des cahiers des charges adaptés au niveau de la conception et un ensemble de paramètres permettant de répondre à ces cahiers des charges peuvent être définis.

Pour mettre en œuvre une telle méthode, il est nécessaire de disposer d'une modélisation adaptée à chaque niveau de détail. Pour cela, les industriels utilisent des simulations pour prédire les performances des systèmes mécaniques. Les modèles mathématiques reflètent plus ou moins le comportement réel sous forme d'un système d'équations à résoudre (cf. figure 1.2). Cette résolution s'effectue soit sous forme analytique soit sous forme numérique de manière approchée en utilisant par exemple la discrétisation. Les principales méthodes numériques sont les méthodes par éléments finis, par différences finies, par volumes finis, par équations intégrales. Les modèles analytiques sont essentiellement utilisés pour valider les choix de définition de l'architecture organique alors que les modèles numériques sont souvent utilisés pour optimiser l'architecture choisie durant la phase de conception ou pour effectuer une validation finale avant de passer au prototype physique [Devalan 2009].

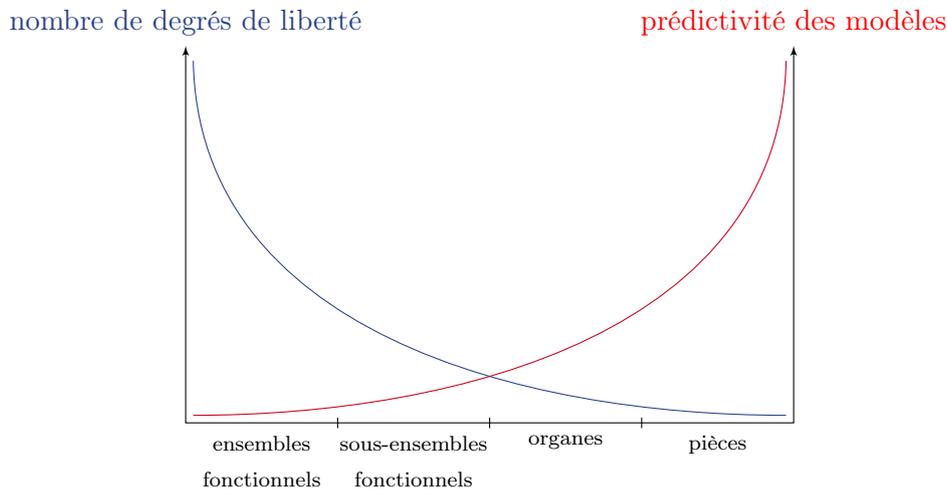


FIGURE 1.2 – Prédictivité des modèles en fonction de l'avancement dans le cycle en V

La phase de conception se divise en deux étapes bien distinctes : l'architecture fonctionnelle et l'architecture organique. La première étape répond à la question « À quoi sert le produit ? », tandis que la vision organique répond à « Comment réalise-t-on en pratique ces fonctions ? ».

La figure 1.3 représente les étapes du déploiement du cahier des charges et distingue les étapes relatives à l'architecture fonctionnelle de celles relatives au dimensionnement organique.

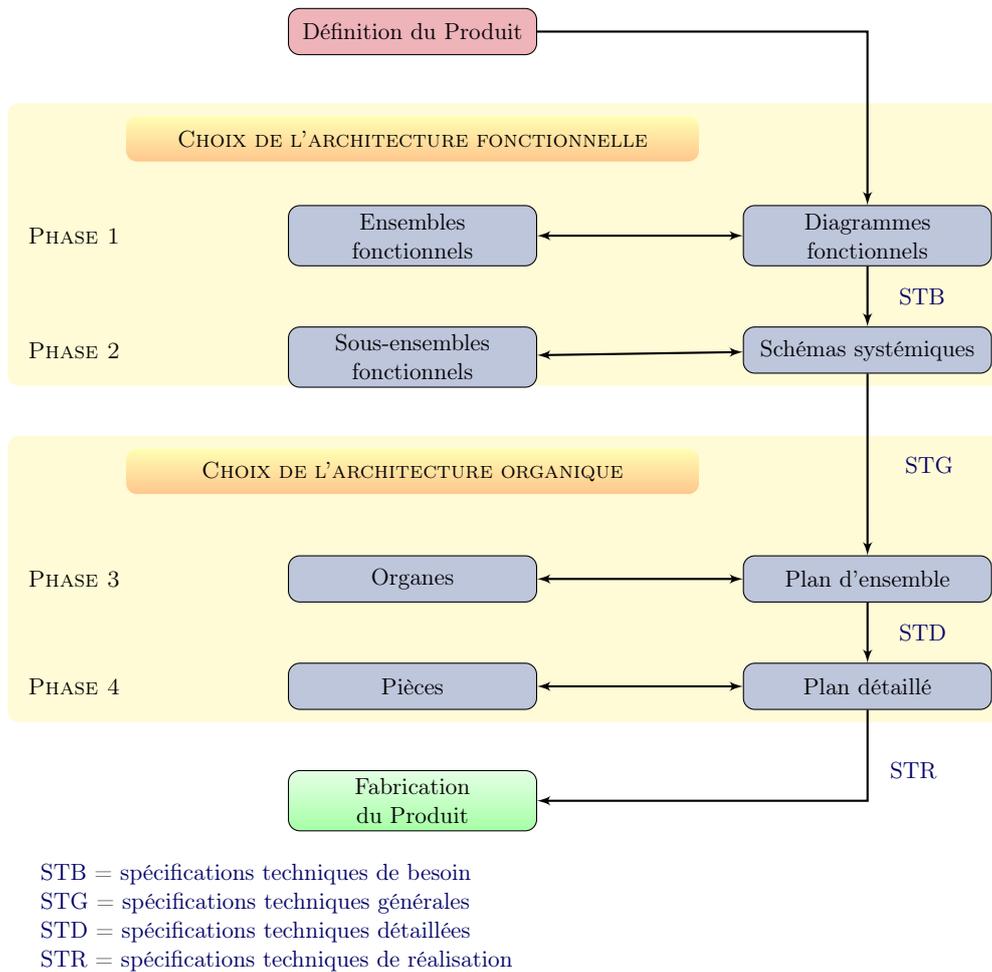


FIGURE 1.3 – Étapes du déploiement du cahier des charges lors de la descente du cycle en V

1.1.1 Architecture fonctionnelle

Ensembles fonctionnels

Dans la première phase de la conception, un diagramme fonctionnel est établi à partir des cahiers des charges du produit. Dans un cadre automobile, ces cahiers des charges incluent globalement les spécifications du cycle de vie, les performances attendues et l'endurance. Ce diagramme fait apparaître les ensembles fonctionnels et les grandeurs caractéristiques sur lesquelles s'appuieront les cahiers des charges. Les spécifications sur les grandeurs d'entrée et de sortie du diagramme constituent les **spécifications techniques de besoin (STB)**.

Pour rendre plus claire cette définition et celles qui suivent, prenons cet exemple simple. Le client exprime son besoin et souhaite « *avoir un véhicule sportif* », ce que l'on traduit entre autres dans un cahier des charges par *une courbe de couple en [N.m], une courbe de puissance en [W], etc.*

Sous-ensembles fonctionnels

La deuxième phase de définition de l'architecture fonctionnelle consiste à décliner le diagramme fonctionnel en schémas systémiques représentant clairement les divers sous-systèmes fonctionnels permettant d'assurer globalement les STB définies précédemment. Le choix des solutions technologiques fonctionnelles et des paramètres fonctionnels de nature physique et géométrique permet de déduire les **spécifications techniques générales (STG)** qui servent alors de cahier des charges lors de la recherche des organes mécaniques et mécatroniques capables d'assurer les fonctions du produit.

Si l'on reprend notre exemple, la courbe de couple va se doter d'un certain nombre de sous-ensembles fonctionnels, chacun ayant une fonction, élément d'une chaîne fonctionnelle aboutissant à la réalisation de la fonction d'usage. En particulier, on s'intéressera par la suite à la *transmission de couple*.

1.1.2 Architecture organique

Organes

La troisième phase consiste à associer à chaque sous-système fonctionnel un organe mécanique ou mécatronique introduisant des paramètres matériels capables d'assurer les STG. Cette relation entre les sous-ensembles fonctionnels et les organes n'est pas bijective : un seul organe peut assurer plusieurs fonctions et une fonction peut être obtenue à l'aide de plusieurs organes.

À ce stade de conception du produit, la géométrie des organes apparaît sous forme simplifiée et des calculs d'ensemble peuvent être effectués pour évaluer les sollicitations sur chaque sous-système. Les niveaux obtenus et les contraintes sur les paramètres géométriques et physiques constituent les **spécifications techniques détaillées (STD)** qui serviront à définir chaque organe.

Si nous reprenons l'exemple précédent, la transmission de couple est répartie sur plusieurs organes : *disque d'embrayage, ressorts, butée*.

Pièces

La quatrième phase de l'architecture organique consiste à dimensionner précisément chaque organe mécanique ou mécatronique, en le scindant la plupart du temps en composants élémentaires appelés pièces. L'analyse des solutions obtenues aboutit aux **spécifications techniques de réalisation (STR)**.

Si l'on isole maintenant le disque d'embrayage, ce dernier est défini par les cotes de base (*diamètre, épaisseur*) et les propriétés du matériau.

1.1.3 Avantages et limites de la méthode

De nombreux acteurs de l'industrie ont adopté cette méthodologie de conception depuis plusieurs années. Cette spécification précise, en briques élémentaires indépendantes, permet une sous-traitance mieux contrôlée. Les différentes équipes peuvent travailler en parallèle. Cela favorise une réduction des cycles de développement ainsi qu'une rationalisation de la conception.

L'utilisation de modèles adaptés à chaque étape de la conception permet de réaliser au mieux les compromis entre les différentes prestations sans avoir à gérer les incertitudes sur le produit,

nombreuses au début du cycle de conception. Dans les phases d'avant-projet, les modèles globaux utilisés sont associés à une vision géométrique simplifiée juste nécessaire pour représenter convenablement les phénomènes physiques. Ces modèles (simplifiés, fonctionnels ou de retour d'expérience), qui ne déterminent que des réponses moyennes d'objets incertains, donnent les bonnes tendances malgré les incertitudes de conception associées aux phases préliminaires.

Enfin, cette organisation de la conception évite un redéveloppement total du produit lorsqu'on ne trouve pas de solution à un niveau de conception : il suffit de remonter au niveau de conception supérieur et de rechercher une autre solution. On redescendra alors au niveau inférieur avec une nouvelle spécification, sans perdre pour autant les informations acquises sur les niveaux de conception supérieurs. Ces bouclages favorisent l'introduction d'innovations : l'échec à atteindre les spécifications en envisageant une technologie donnée pousse à en envisager une autre.

L'aspect itératif de la recherche de solutions peut cependant poser des problèmes de convergence vers une solution satisfaisant le cahier des charges à cause des couplages entre disciplines. En effet, les bouclages de conception peuvent être coûteux en temps et en argent. C'est pourquoi le découpage du produit en sous-systèmes indépendants a été largement étudié, comme le montre la partie suivante sur les stratégies d'optimisation multi-disciplinaires.

1.2 Familles de paramétrage de forme

Le recours à l'optimisation de forme pour la conception de produits est motivé par le souci de réduire les temps de développement et de fournir une solution performante et robuste. Les approches couramment utilisées par les industriels sont [Saitou 2005, Roy 2008] :

- o l'optimisation paramétrique basée :
 1. sur la CAO (conception assistée par ordinateur) ;
 2. sur le maillage : cette technique repose sur la déformation d'un maillage en regroupant un ensemble de nœuds dans une boîte de morphing ; chacune des boîtes est un paramètre, la position des nœuds est alors modifiée de manière globale.
- o l'optimisation isogéométrique, qui exploite le modèle géométrique comme support de calcul en se basant sur les fonctions de base (B-splines, NURBS...) fournies par les logiciels CAO ;
- o l'optimisation géométrique, qui fait varier indépendamment chaque nœud de la frontière de la pièce ;
- o l'optimisation topologique, qui consiste à trouver la répartition de matière idéale dans un volume donné soumis à des contraintes.

Ces méthodes seront détaillées par la suite. Elles sont dédiées à la conception de pièces, elles interviennent donc assez tardivement dans le cycle en V. Comme l'indique la figure 1.2, ces méthodes opèrent dans un environnement contraint où il demeure difficile d'apporter des solutions innovantes. [Roy 2008] propose une comparaison de ces méthodes de paramétrage, récapitulée et enrichie dans le tableau 1.1. Une lecture rapide de ce tableau nous montre qu'il n'existe pas de méthode universelle répondant à toutes les attentes du concepteur.

Ces méthodes sont industrialisées et disponibles au travers de logiciels comme OptiStruct[®] (Altair), RODIN[®] (projet FUI), ANSA[®] (BETA CAE Systems), CATIA[®] (Dassault Systèmes)...

1.2. Familles de paramétrage de forme

Type d'optimisation	Paramètres	Nombre de paramètres	Dépendance au concepteur	Capacité à respecter des procédés de fabrication	Prérequis
Paramétrique	cotes CAO, morphing	$10^1, 10^2$	très forte	++	paramétrisation & automatisation
Isogéométrique	points de contrôle	$10^2, 10^3$	moyenne	–	géométrie définie par carreaux de splines & solveur spécifique
Géométrique	position des nœuds	$10^6, 10^7$	faible	–	calcul des gradients & déformation de maillage
Topologique	densité/porosité des éléments	$10^6, 10^7$	faible	–	calcul des gradients

Tableau 1.1 – Comparaison entre les différents types de paramétrage

1.2.1 Paramètres métier pilotés par CAO

La définition 3D de la pièce est réalisée à l'aide d'un logiciel CAO. Les principaux systèmes sont CATIA[®] édité par Dassault Systèmes, NX[®] publié par Siemens PLM Software et Creo[®] fait par PTC. Les *paramètres métier*^a sont introduits dans la CAO. Il est ensuite aisé de faire varier la géométrie via une table de paramétrage [Suh 2002]. Il est également possible de jouer sur la topologie d'une pièce en utilisant des paramètres qualitatifs de type présence/absence ou de substituer une pièce par une autre dans un système. L'avantage d'une telle méthode réside dans sa capacité à respecter des procédés de fabrication car ces derniers peuvent être directement mis en œuvre au travers du paramétrage de la géométrie.

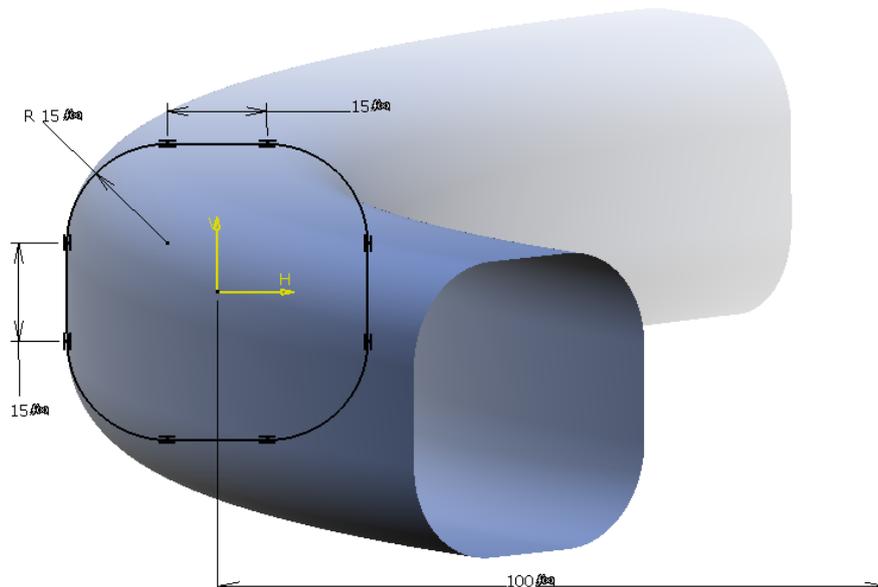


FIGURE 1.4 – Exemple d'une CAO

a. Les *paramètres métier* sont des paramètres géométriques élémentaires (épaisseur, congé, longueur, angle...) ou bien des paramètres issus de l'expérience (galbe du pavillon, resserrée arrière...).

La flexibilité d'une géométrie dépend donc directement du nombre de paramètres : plus le nombre de paramètres augmente, plus l'espace de conception balayé sera important. On est donc tenté de multiplier le nombre de paramètres afin d'explorer l'espace de conception le plus large possible. Toutefois, l'augmentation du nombre de paramètres se fait au détriment de la robustesse du modèle paramétré [Saitou 2005, Roy 2008]. Le taux de reconstruction des géométries est, en général, intimement lié au nombre de paramètres. Parmi les sources possibles d'échec lors de la génération des géométries, on peut lister l'incompatibilité entre les règles de conception, les bogues intrinsèques aux logiciels, etc.

Sur un tout autre registre, l'espace à explorer croît avec le nombre de degrés de liberté, augmentant d'autant plus le nombre de calculs numériques nécessaires lors de la recherche d'une forme optimale avec un plan d'expériences. Au niveau industriel, les boucles d'optimisation s'appuyant sur un modèle CAO utilisent des plans d'expériences et une recherche de l'optimum par surface de réponse. Pour mettre en place une telle boucle, il faut donc dans un premier temps créer une géométrie qui permette d'explorer l'espace de conception le plus vaste possible tout en maîtrisant le taux de reconstruction du modèle. Cet espace de conception est ensuite exploré à l'aide d'un plan d'expériences. Il est donc nécessaire de développer des scripts permettant d'automatiser toutes les étapes entre la génération des géométries et le post-traitement des résultats. Le plan d'expériences permet à l'issue des calculs de créer la surface de réponse, laquelle va nous permettre de trouver des géométries candidates dans notre recherche de la forme optimale (voir la figure 1.5).

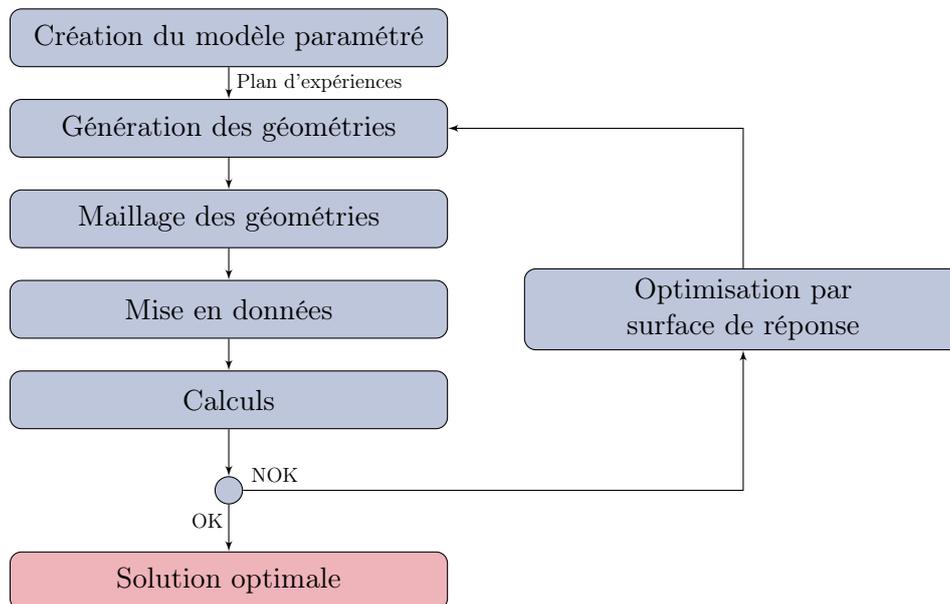


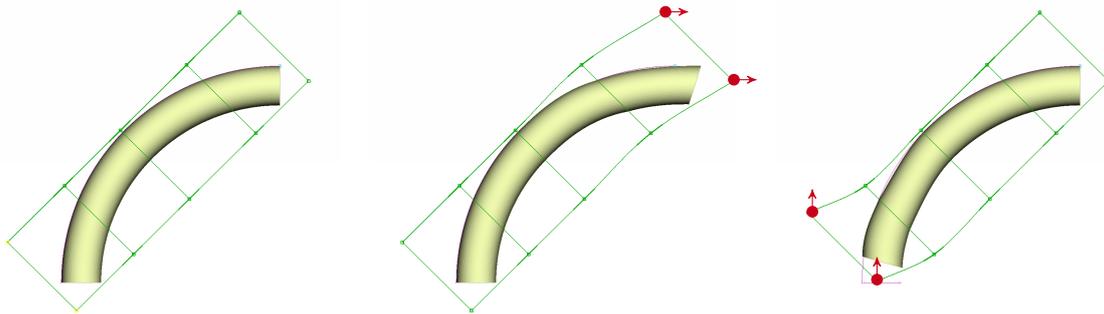
FIGURE 1.5 – Exemple d'une boucle d'optimisation paramétrique

L'intérêt principal de cette méthode repose sur le fait qu'elle n'est pas intrusive par rapport aux codes de calcul. L'ingénieur peut alors définir une boucle d'optimisation à l'aide des logiciels auxquels il est habitué (voir [Brujic 2010]). Un autre aspect important de cette méthode est que le nombre de paramètres servant à l'optimisation est défini par le concepteur qui décide donc d'introduire plus ou moins de complexité dans la construction de la géométrie. En contrepartie, plus le nombre de paramètres sera limité et plus le côté innovant de la solution sera faible. Cette

solution limite considérablement la variété des formes possibles (ou admissibles). Ce dernier point est lié au fait que la conception se base sur une définition de détail de la pièce ou de l'organe. Un autre point négatif de la méthode est le coût élevé de la création d'une CAO paramétrée et des scripts d'automatisation nécessaires à l'élaboration d'une boucle de calcul, notamment avec une étape de maillage qui peut s'avérer particulièrement délicate.

1.2.2 Paramètres métier pilotés par morphing

Plutôt que de modifier le modèle CAO puis de le mailler, il peut être intéressant de paramétrer directement le maillage et de le déformer de manière globale [Yang 1992]. Une variation de la forme est générée en modifiant l'ancrage géométrique de la forme initiale tout en conservant sa topologie fixée. L'intérêt de cette méthode est de limiter le nombre de paramètres, d'éviter de paramétrer un modèle CAO et de réduire le travail d'automatisation de la chaîne de calcul. Les techniques de morphing visent à préserver la qualité des éléments autant que possible. Pourtant, cette approche reste beaucoup moins permissive que celle basée sur une CAO car on ne parvient pas à préserver la qualité de maillage. Staten et ses collègues [Staten 2011] ont récemment proposé et évalué plusieurs techniques de morphing qu'ils comparent dans le domaine de la performance de calcul et de la qualité des éléments sur différents maillages tétraédriques et hexaédriques.



(a) Forme initiale avec 4 boîtes de morphing (b) Déplacement de 30 mm des deux nœuds supérieurs selon X (c) Déplacement de 30 mm des deux nœuds inférieurs selon Y

FIGURE 1.6 – Illustration d'un morphing

En pratique, l'utilisateur définit des boîtes de morphing. Chaque boîte contrôle le déplacement des nœuds du maillage à proportion de la distance qui les sépare des points de contrôle. La difficulté de ce paramétrage est de trouver le bon compromis entre le nombre de boîtes définies et la finesse de modification souhaitée.

1.2.3 Paramétrage dédié à l'optimisation isogéométrique

L'analyse isogéométrique a été récemment introduite par Hughes [Hughes 2005]. C'est une méthode des éléments finis d'ordre élevé (*high-order FEM*), c'est-à-dire qu'il est nécessaire d'utiliser des polynômes rationnels au minimum quadratiques pour représenter des cercles, des cylindres ou des sphères par exemple. L'idée de l'analyse isogéométrique est de rapprocher l'analyse par éléments finis de la modélisation CAO : les *patches CAO* servent d'*éléments* et les *splines* les

définissant (B-splines, NURBS. . .) servent de *fonctions d'interpolation*. Cela permet de respecter exactement la géométrie et d'éliminer par conséquent les erreurs dues aux approximations de la forme. Cette technique a l'avantage de ne pas faire appel à un outil de maillage. Toutefois, une phase d'enrichissement de la géométrie (*raffinement h-p-k* [Cottrell 2009]) est nécessaire pour obtenir une bonne convergence des calculs (voir la figure 1.7).

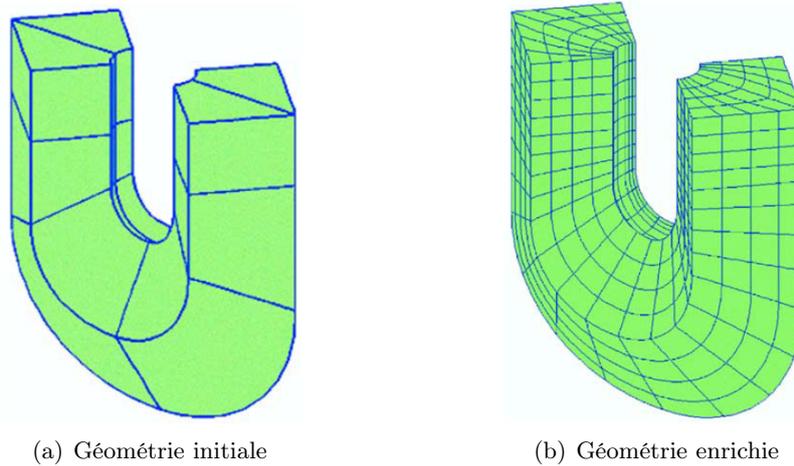


FIGURE 1.7 – Enrichissement d'une géométrie par raffinement h-p-k (images extraites de [Hughes 2005])

L'analyse isogéométrique nécessite donc de définir notre CAO par des carreaux de splines, ce qui est rarement le cas dans l'ingénierie actuelle. Qui plus est, bien que les NURBS soient largement développées dans les systèmes CAO, il est rare d'avoir accès à la définition de la géométrie au sein des logiciels sauf à utiliser des formats neutres tels que le STEP ou l'IGES. En outre, cette méthode nécessite une refonte des codes de calcul. Elle est, par exemple, en cours d'implémentation par [de Falco 2011] dans GeoPDEs[®] et dans le code industriel LS-DYNA[®] de Siemens.

Enfin, on notera qu'une telle approche permet de lier facilement les paramètres CAO au calcul dès lors qu'on utilise les positions des points de contrôle des splines comme paramètres [du Cauzé de Nazelle 2013]. Toutefois, si on utilise des paramètres métier, ce lien devient très difficile du fait qu'on ne connaît pas la sensibilité des splines par rapport à ces paramètres. C'est d'autant plus vrai si des patches apparaissent lorsqu'un paramètre métier varie.

1.2.4 Paramétrage dédié à l'optimisation topologique

L'optimisation topologique consiste à déterminer le réseau optimal de matière dans un volume donné. Ce volume doit respecter l'ensemble des contraintes d'architecture. Il est fourni sous la forme d'un maillage. Chaque élément de ce maillage est un degré de liberté d'optimisation. Il peut donc y en avoir plusieurs millions. Pour ce faire, on va piloter la densité de matière par élément (0 : matériau vide ; 1 : matériau plein). C'est un problème combinatoire impossible à gérer dans de telles dimensions. Aussi, une idée simple a été proposée : relaxer le problème en pilotant continûment cette densité [Bendsoe 2003]. Grâce à cette astuce, on peut calculer un gradient et donc résoudre ces grands problèmes avec des techniques classiques d'optimisation (voir la figure 1.8).

Pour pallier les problèmes de densités intermédiaires qui rendent difficiles l'interprétation de la solution, des approches différentes sont développées. Le projet FUI RODIN[®] utilise par exemple l'approche des lignes de niveau (*level set*) [Dapogny 2013, Michailidis 2014]. De façon générale, la solution d'une optimisation topologique sera capturée avec autant de précision que le maillage initial est fin.

Ces méthodes ont un gros potentiel : elles ne nécessitent aucun paramétrage mais elles requièrent de savoir calculer un gradient et de mathématiser toutes les contraintes du cahier des charges. À ce titre, les contraintes de *design* ne sont pas aisément gérables.

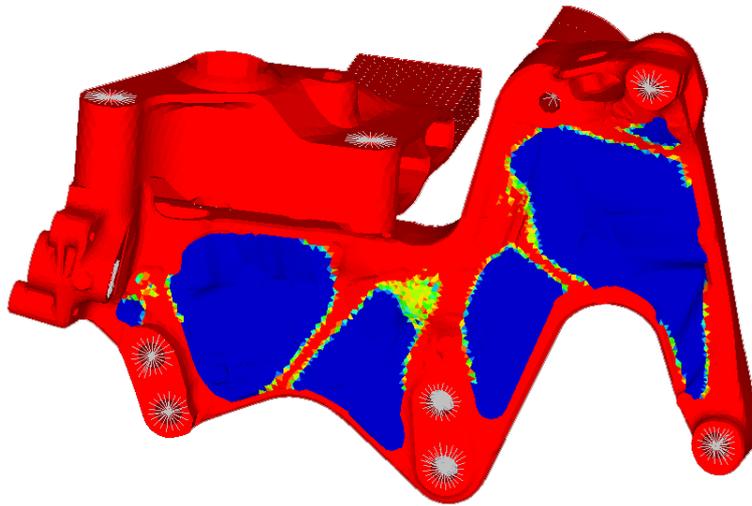


FIGURE 1.8 – Illustration d'une optimisation topologique : les zones rouges indiquent la présence de matière (densité de 1) et les zones bleues les trous (densité nulle). Le reste est constitué de densités intermédiaires, ce qui n'a pas de réelles significations physiques.

1.2.5 Paramétrage dédié à l'optimisation géométrique

L'optimisation géométrique consiste à modifier le maillage à chaque itération du processus d'optimisation (cf. figure 1.9). Cela permet de toujours avoir une description fine et exacte de la forme et de calculer proprement toutes les grandeurs mécaniques.

1.2.6 Variante d'optimisation topologique dédiée à la CFD

Il existe une déclinaison de l'optimisation topologique dédiée à la mécanique des fluides. Cette approche sans gradient est développée dans le logiciel TOSCA Fluid[®] (édité par DS SIMULIA[®]) qui traite de problèmes d'écoulement fermé. Elle est basée sur la prévention d'un retour d'écoulement sur une base de cellule à cellule et utilise les résultats d'un calcul CFD. L'objectif est de prévenir les zones de recirculation du domaine fluide [Moos 2004]. Cette méthode n'est pas basée sur un calcul de sensibilité et n'a pas de formulation mathématique rigoureuse. Les critères actuellement gérés sont la perte de charge et l'homogénéisation d'un profil de vitesse.

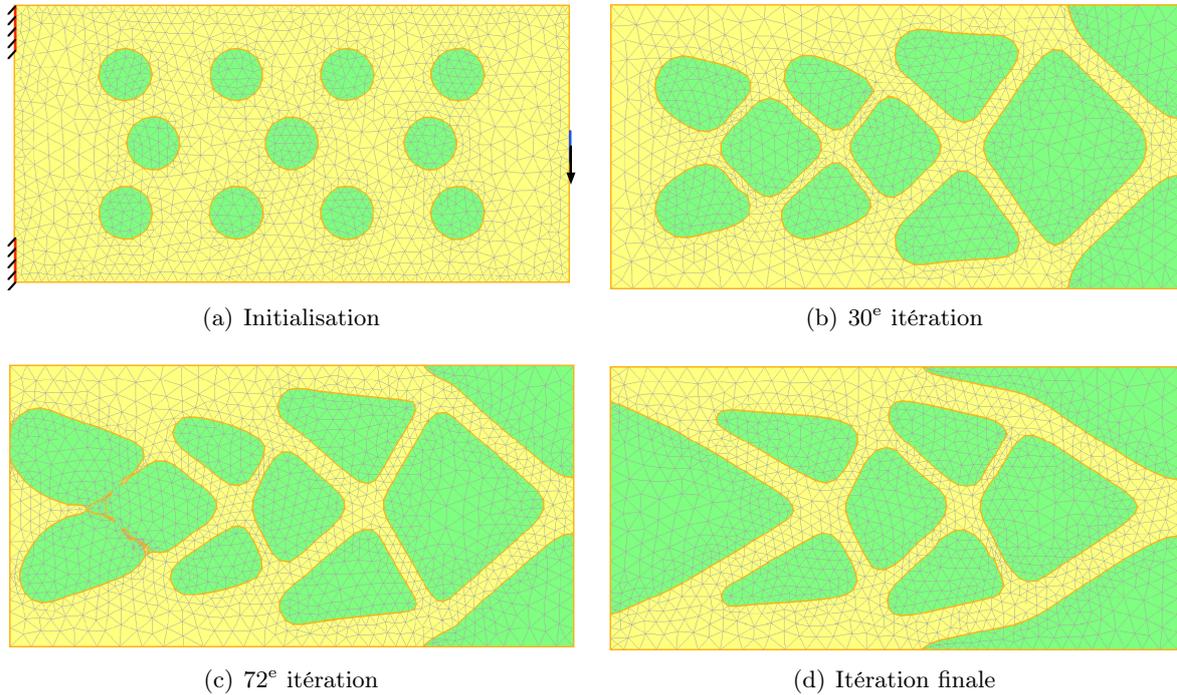


FIGURE 1.9 – Illustration d'une optimisation géométrique d'une console (images extraites de la thèse [Dapogny 2013])

Déroulement de la méthode

Cette méthode se limite à des problèmes d'écoulement dans un conduit en régime permanent. Chaque cellule de l'espace de conception est considérée comme une variable d'optimisation. Le solveur modifie l'espace de conception de façon à ce que le conduit ne possède aucun refoulement, *i.e.* des zones de recirculation non voulues où la dissipation d'énergie entraîne une chute de pression et où les tourbillons locaux peuvent créer du bruit (cf. figure 1.10).

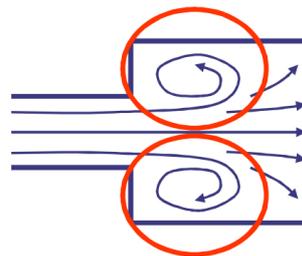


FIGURE 1.10 – Zones de refoulement

Le principe est d'empêcher l'écoulement dans une cellule qui appartient à une zone de recirculation : la cellule est alors sédimentée. Pour cela, les trajectoires primaires des particules sont déterminées entre chaque entrée et sortie. Elles passent toujours par le centre de la/des sortie(s) (figure 1.11(a)). Puis ces trajectoires primaires servent de référence pour les directions d'écoulement. Ainsi, une direction d'écoulement est définie comme référence pour chaque cellule (figure 1.11(b)). Pendant l'optimisation, l'écoulement évolue et la direction de l'écoulement ac-

tuel de chaque cellule est calculée (figure 1.11(c)). La déviation entre la direction de l'écoulement actuel et celle de l'écoulement de référence contrôle la sédimentation de la cellule qui est basée sur un critère angulaire. L'angle critique est défini par une tolérance de refoulement (figure 1.11(d)) comprise entre 0 et 1. Une valeur proche de 0 signifie que l'écart entre la direction de l'écoulement actuel et celle de l'écoulement de référence est infime.

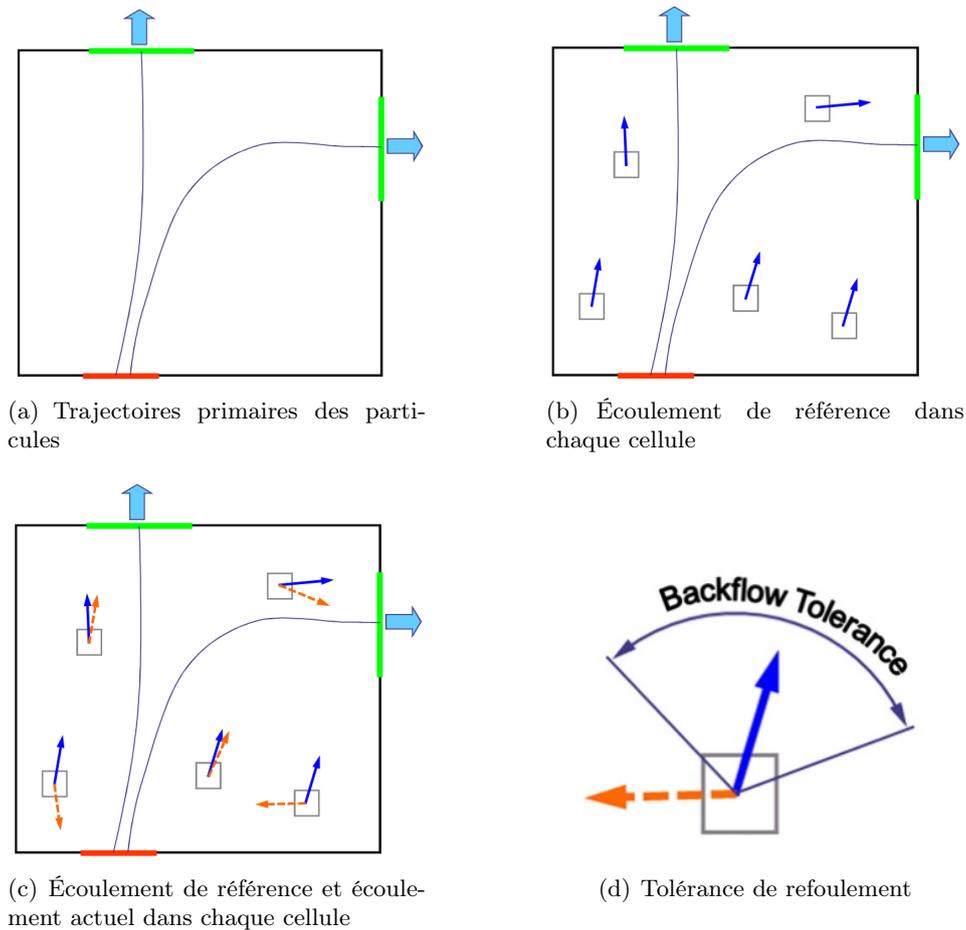


FIGURE 1.11 – Sédimentation des cellules et tolérance de refoulement

L'utilisateur peut agir sur la tolérance de refoulement et sur le niveau de coupure de la sédimentation des cellules. Le premier paramètre définit la quantité de retour autorisée par l'optimiseur. Le second influence fortement la forme (figure 1.12), comme mentionné dans le paragraphe 1.2.4 (p. 12).

Avantages et inconvénients

L'avantage de cette méthode est de permettre de trouver de nouvelles formes [Moos 2004]. Toutefois, elle souffre des mêmes maux que l'optimisation topologique avec gradient. En effet, on peut rappeler notamment que les frontières ne sont pas connues au cours de l'optimisation, ce qui est particulièrement dommageable car il n'est pas possible de définir de couches limites, ce qui entraîne des erreurs de simulation non négligeables. D'autre part, ce type d'optimisation traite

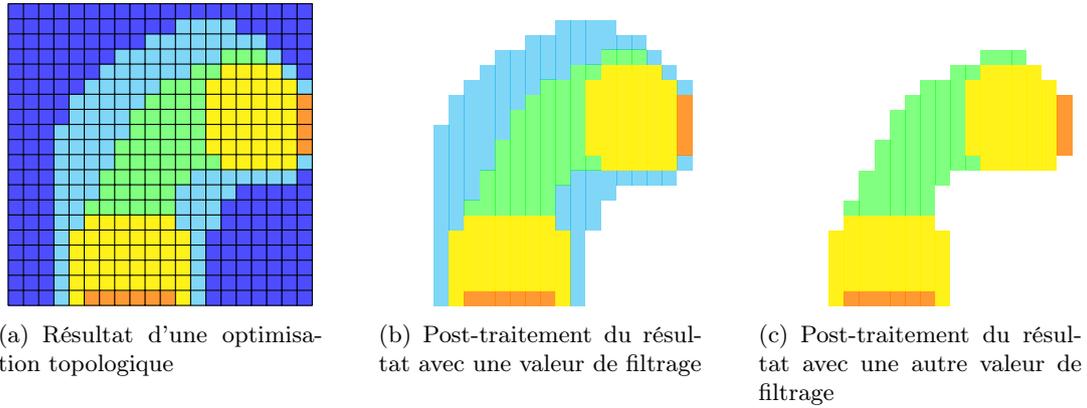


FIGURE 1.12 – Influence du niveau de coupure

seulement à l'heure actuelle des problèmes incompressibles, isothermes et en régime stationnaire, dont le nombre de Reynolds doit rester petit. Enfin, le retour dans un environnement CAO reste un point délicat.

1.3 Méthodes d'optimisation (*gradient-based* ou *gradient-free*)

Au fil du manuscrit, différentes méthodologies d'optimisation sont utilisées et sont donc présentées dans cette section. Un problème d'optimisation type est de la forme [Allaire 2007] :

$$\begin{cases} \min_{x \in \mathcal{G}_{\text{ad}}} [J_1(x), \dots, J_m(x)] \\ \text{s.t.} \begin{cases} g_j(x) \leq 0, & j = 1, \dots, n_i \\ h_l(x) = 0, & l = 1, \dots, n_e \end{cases} \end{cases} \quad (1.1)$$

où :

- m est le nombre de fonctions objectif. Si $m = 1$, c'est un problème mono-objectif. Si $m > 1$, c'est un problème multi-objectif ;
- n est le nombre de variables d'optimisation ;
- n_i est le nombre de contraintes d'inégalité ;
- n_e est le nombre de contraintes d'égalité ;
- $\mathcal{G}_{\text{ad}} \subseteq \mathbb{R}^n$ représente l'*espace de conception*, c'est-à-dire l'espace alloué pour concevoir la pièce. D'un point de vue formel, \mathcal{G}_{ad} est l'espace des fonctions dans lequel les variables d'optimisation x peuvent prendre leurs valeurs. D'après la section 1.2 (p. 8), ces variables peuvent être :
 - des *paramètres métier* de type CAO ou morphing : rayons, épaisseurs, longueurs, angles, etc. ;
 - des *paramètres géométriques* : position des nœuds du maillage ;
 - des *paramètres topologiques* : densité de matière ou porosité des éléments.
- $J: \mathcal{G}_{\text{ad}} \rightarrow \mathbb{R}^m$ représente les *fonctions objectif* (appelées aussi *fonctions coût*), issu par exemple de la résolution d'un problème mécanique (déplacement, fréquence propre, coefficient de trainée aérodynamique...).

- $g: \mathcal{G}_{\text{ad}} \rightarrow \mathbb{R}^{n_i}$ représente les *contraintes d'inégalité* (volume, von Mises...);
- $h: \mathcal{G}_{\text{ad}} \rightarrow \mathbb{R}^{n_e}$ représente les *contraintes d'égalité*.

Ci-après, deux exemples de problème de conception optimale que l'on retrouvera plus tard dans le troisième chapitre (p. 85).

Problème en raideur statique [Allaire 2003]

Dans cet exemple, on souhaite maximiser la raideur que l'on voit sous l'angle du déplacement du nœud chargé. Pour cela, on pose $J(u(x)) = \|u_m\|$ avec u_m le déplacement du nœud maître sur lequel est appliquée la force. On se donne un domaine borné $\Omega \subset \mathbb{R}^3$ dont le bord est constitué de trois frontières $\partial\Omega = \Gamma \cup \Gamma_D \cup \Gamma_N$ avec des conditions limites de Dirichlet sur Γ_D et de Neumann sur Γ_N (cf. figure 1.13). On a :

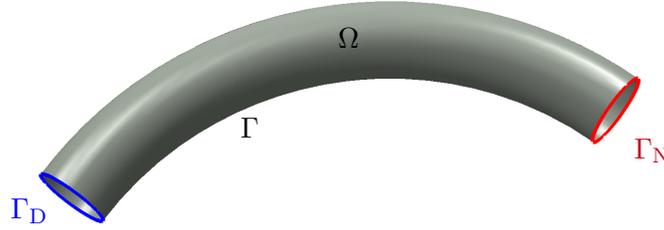


FIGURE 1.13 – Exemple de mise en données d'un problème d'optimisation de forme

$$\begin{cases} \min_x J(u(x)) \\ \text{s.t. } \underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall i = 1, \dots, n \end{cases} \quad (1.2)$$

avec $u(x)$ solution du problème d'élasticité linéarisée suivant :

$$\begin{cases} -\operatorname{div} \sigma(u) = f & \text{dans } \Omega \\ u = 0 & \text{sur } \Gamma_D \\ \sigma(u) \cdot n = g & \text{sur } \Gamma_N \\ \sigma(u) \cdot n = 0 & \text{sur } \Gamma \end{cases} \quad (1.3)$$

f est le vecteur des forces volumiques, g le vecteur des forces surfaciques, n le vecteur unitaire normal à $\partial\Omega$ et σ le tenseur des contraintes qui suit la loi de Hooke.

Problème en comportement vibratoire [Allaire 2004]

On souhaite ici maximiser la pulsation ω_1 du premier mode propre, ce qui équivaut à minimiser $J(\omega(x)) = -\omega_1^2$. On a :

$$\begin{cases} \min_x J(\omega(x)) \\ \text{s.t. } \underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall i = 1, \dots, n \end{cases} \quad (1.4)$$

On note $\omega(x) \geq 0$ la pulsation propre et $\phi(x)$ le mode associé (c'est-à-dire le champ de déplacement correspondant sur Ω) qui sont solutions du problème suivant :

$$\begin{cases} -\operatorname{div} \sigma(\phi) = \omega^2 \rho \phi & \text{sur } \Omega \\ \phi = 0 & \text{sur } \Gamma_D \\ \sigma(\phi) n = 0 & \text{sur } \Gamma_N \cup \Gamma \end{cases} \quad (1.5)$$

avec ρ la densité volumique du matériau et σ le tenseur des contraintes qui suit la loi de Hooke. (3.27) admet une infinité de solutions $(\omega_k, \phi_k)_{k \geq 1}$ que l'on classe dans l'ordre croissant des pulsations.

Pour résoudre ces problèmes d'optimisation, il existe principalement deux approches : l'une est basée sur les plans d'expériences et ne requiert aucune spécificité, ce qui la rend assez universelle ; l'autre s'appuie sur la connaissance des gradients et exige *a contrario* d'être en mesure de les calculer.

1.3.1 Méthodes d'optimisation par plans d'expériences et surfaces de réponse (*gradient-free*)

Les méthodes d'optimisation par plans d'expériences consistent à organiser l'exploration d'un espace de conception. Pour cela, on crible une première fois l'espace puis on évalue la solution. On obtient ainsi une valeur par critère qui va servir à construire un modèle statistique^b sur lequel on cherchera un optimum virtuel que l'on validera par calcul. Le tout est schématisé dans la figure 1.14.

Les grandes étapes [Benoist 1994] du déroulement de cette méthode d'optimisation sont :

1. formalisation du problème : il s'agit de décrire les objectifs et retombées attendues du plan d'expériences d'une part, les entrées et sorties de l'optimisation d'autre part,
2. construction du plan d'expériences : il faut définir le nombre d'essais nécessaires pour l'étude des paramètres,
3. réalisation du plan d'expériences,
4. analyse des résultats d'essais : cette étape consiste à ajuster un modèle de régression expliquant les influences,
5. interprétation et validation des résultats d'essais : l'objectif est de faire la synthèse des connaissances acquises et de valider les conclusions.

Les avantages de cette démarche sont nombreux : d'une part, elle n'est pas intrusive vis-à-vis des logiciels utilisés ; d'autre part, elle permet d'estimer des fonctions non différentiables par rapport aux paramètres (comme pour les simulations de crash). En contrepartie, le choix du modèle de régression se doit d'être cohérent avec la physique mise en jeu. La convergence de ce type de méthodes peut s'avérer longue, toutefois l'expérience montre que, dans les cas usuels, un nombre raisonnable d'essais permet d'améliorer la fonction coût de manière significative.

En fonction du problème considéré, il existe une certaine latitude pour gagner en performance et en efficacité dans l'exécution de la méthode. Par exemple, si l'on sait par avance le degré de non-linéarité d'une fonction, on peut intelligemment choisir le modèle statistique (degré 1 d'un polynôme, krigeage...).

1.3.1.1 Plans d'expériences

Les plans d'expériences n'exigent aucune connaissance du problème étudié. À ce titre, leur pratique est plutôt populaire dans les milieux industriels, de la conception à la fabrication. Ils sont

b. Le modèle statistique est connu aussi sous les noms de *métamodèle* ou *surface de réponse*.

1.3. Méthodes d'optimisation (*gradient-based* ou *gradient-free*)

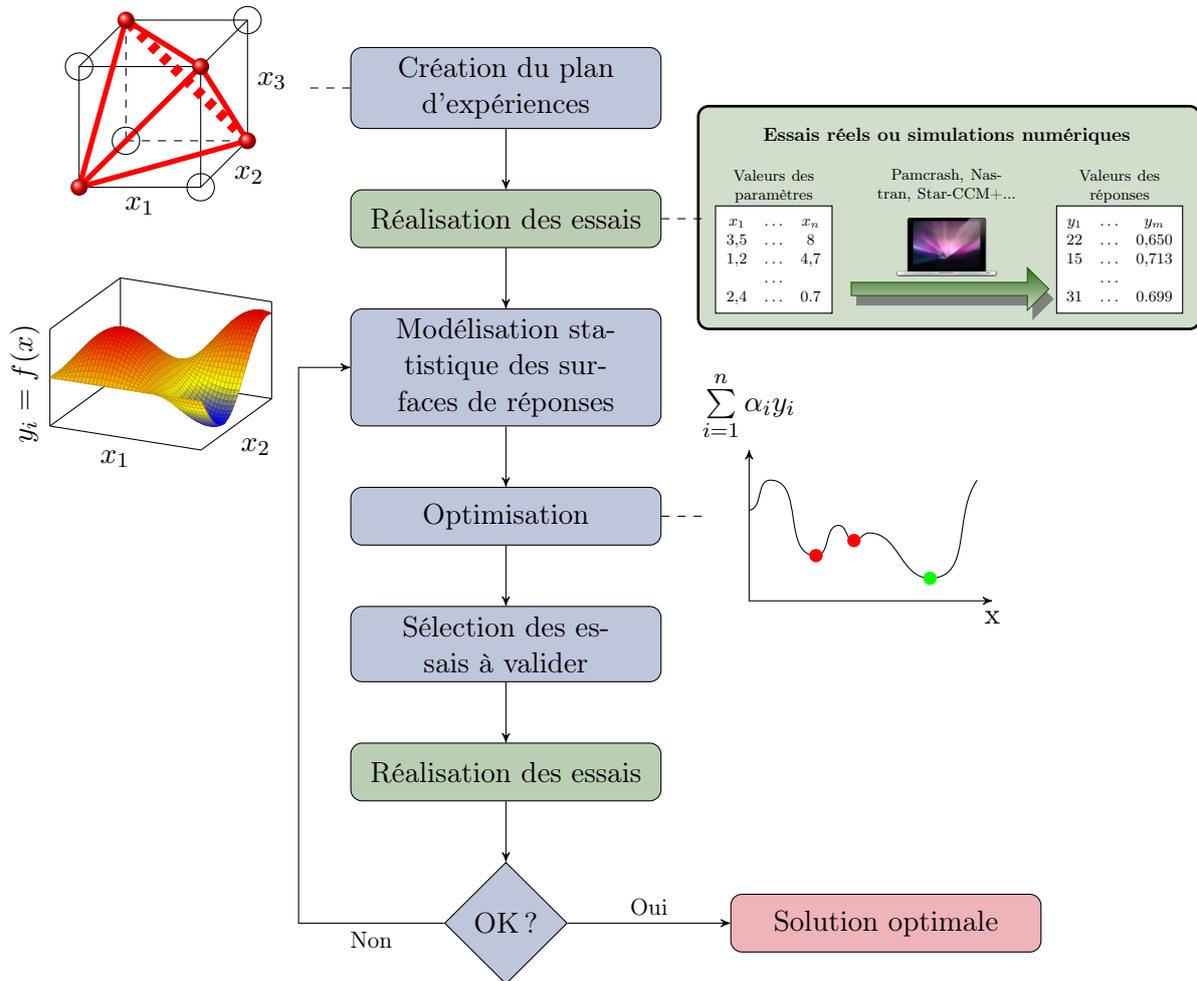


FIGURE 1.14 – Schéma représentant une méthode d'optimisation par plan d'expériences

couramment utilisés sur des phénomènes de type « boîte noire » dont on cherche à comprendre le fonctionnement et optimiser les performances. Ils permettent de planifier de manière rationnelle un ensemble d'expériences afin d'en tirer le maximum de renseignements avec un minimum d'essais physiques (ou de calculs). Cette organisation des essais permet également d'obtenir la meilleure précision possible sur la modélisation des résultats.

Pour avoir un plan, on peut soit le tirer d'un catalogue de tables déjà construites soit le construire *ad hoc* (plans D-optimaux, plans G-optimaux, etc.), ce qui a un coût. On renvoie le lecteur vers les ouvrages spécialisés comme [Benoist 1994, Goupy 1999]. Parmi les plans disponibles, on peut citer les plans complets, les plans fractionnaires^c, les plans de Box-Behnken... Tous ces plans sont dédiés à la construction d'un modèle polynomial défini *a priori* (c'est-à-dire son degré, le type de monômes, etc.). Les plans hypercubes latins^d sont moins dédiés à un modèle mais n'ont d'intérêt que si l'évaluation des essais est rapide et que le phénomène est non linéaire.

Les plans LHS et les plans optimaux sont présentés plus en détail dans les annexes B.1

c. Les plans fractionnaires sont également appelés tables de Taguchi.

d. Par la suite, les plans hypercubes latins seront désignés par *plans LHS* (selon le sigle d'origine anglo-saxonne *Latin hypercube sampling*)

(p. 125) et B.2 (p. 129).

1.3.1.2 Surfaces de réponse

Les surfaces de réponse sont des approximations de la fonction réelle. Pour construire une surface, il existe différentes stratégies. De nombreux outils sont disponibles dans Alternova[®], logiciel développé en partenariat entre Renault et EURODECISION. Alternova[®] est une plateforme d'optimisation multi-objectif pour le dimensionnement de systèmes complexes, qui permet d'exploiter au mieux différents modèles statistiques grâce au *mélange d'experts* initialement développé par [Jacobs 1991].

Les différentes méthodes qui existent sont classées en deux catégories : une approche est dite non paramétrique si l'on prend une classe de fonctions générales (par exemple, le krigeage), par opposition au fait que l'on choisisse une fonction. Dans ce dernier cas, on parle d'approche paramétrique (on subodore alors l'allure de la solution ce qui entraîne un risque de se tromper de fonction). On détaille dans les paragraphes qui suivent les méthodes les plus usuellement utilisées dans l'industrie. Les modèles polynomiaux sont largement employés pour leur simplicité, les modèles additifs MARS et polyMARS sont quant à eux particulièrement adaptés à la grande dimension. Enfin, le krigeage présente l'avantage d'être une méthode interpolante aboutissant à la construction d'un modèle probabiliste.

Modèles polynomiaux

Les modèles polynomiaux sont largement répandus dans l'industrie. La démarche consiste à construire dans un premier temps la matrice X du plan pour le modèle polynomial choisi. On détermine ensuite le vecteur y des réponses du système associées aux essais du plan d'expériences. On cherche alors la matrice β des estimateurs (équation (1.6)) qui minimise l'erreur de modélisation commise ε :

$$y = X\beta + \varepsilon \tag{1.6}$$

Augmenter le degré du modèle polynomial permet de capter des interactions entre certains facteurs et de modéliser des phénomènes physiques complexes. Toutefois, cela nécessite la réalisation d'un plus grand nombre d'essais pour évaluer les coefficients du polynôme. En pratique, il est commode d'utiliser des modèles polynomiaux d'ordre 2 :

$$y = \beta_0 + \sum_i \beta_i x_i + \sum_i \beta_i x_i^2 + \sum_{i < j} \beta_{ij} x_i x_j + \varepsilon \tag{1.7}$$

Le nombre de coefficients à estimer pour un modèle quadratique – en notant p le nombre de paramètres – est donné par l'équation (1.8). Si l'on se donne 10 paramètres, cela fait donc 66 coefficients à estimer.

$$\underbrace{1}_{\text{constante}} + \underbrace{2}_{\text{termes quadratiques}} \cdot \underbrace{p}_{\text{termes linéaires}} + \underbrace{\frac{p \cdot (p - 1)}{2}}_{\text{interactions}} \tag{1.8}$$

MARS - polyMARS

La régression multivariée par spline adaptative (en anglais **MARS** pour **multivariate adaptive regression spline**) [Friedman 1991] utilise une somme de fonctions de base $B_j(x)$ (cf. équation(1.9)) qui prennent la forme d'une fonction charnière (voir la figure 1.15) ou d'un produit de fonctions charnières.

$$y = \beta_0 + \sum_{j=1}^k \beta_j B_j(x) \tag{1.9}$$

Ces modèles permettent essentiellement de créer une cassure pour tenir compte de la non-linéarité des données. Ainsi, à partir de modèles linéaires, il est possible de modéliser des fonctions ayant un comportement non linéaire. Une extension de **MARS** proposée par [Kooperberg 1997] permet de construire des modèles **polyMARS** (**polychotomous regression based on MARS**). Ces modèles sont utilisés dans le cas de réponses multiples et sont davantage adaptés à la grande dimension que **MARS**.

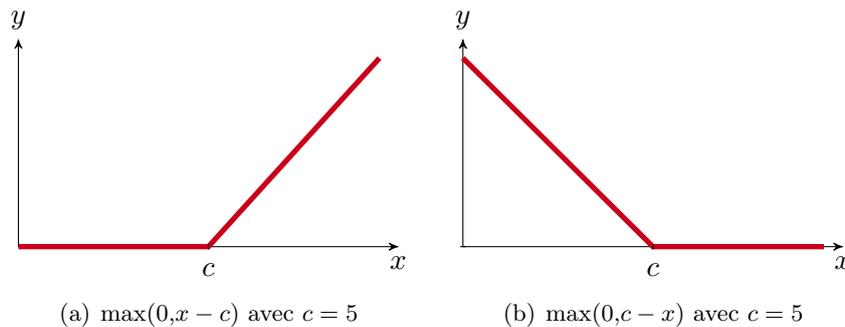


FIGURE 1.15 – Fonctions charnières des modèles MARS

Krigeage

Le *krigeage* est une méthode intermédiaire [Dupuy 2007] entre :

- des méthodes rigides comme la régression polynomiale pour lesquelles la famille de fonctions utilisées pour construire la surface est imposée en amont ;
- des méthodes souples comme les modèles additifs ou **polyMARS** où l'estimation repose sur la construction de splines dans chaque direction.

C'est une méthode de prédiction reposant sur un modèle probabiliste (généralement gaussien) qui est utilisé entre autres pour interpoler des données et décrire les incertitudes sur la fonction en d'autres points. Il porte le nom de son précurseur – l'ingénieur minier sud-africain D.G. Krige – et a été formalisé mathématiquement par George Matheron du Centre de Géostatistiques de l'École des Mines de Paris [Matheron 1963].

Le krigeage est un problème classique d'estimation d'une fonction déterministe continue et coûteuse numériquement $f(x)$ de $S \in \mathbb{R}^d \rightarrow \mathbb{R}$. La fonction inconnue est modélisée par un processus gaussien de fonction moyenne $\mathbb{E}(F(x)) = \mu$, supposée constante, et de fonction de covariance $k(x,y) = \text{Cov}(F(x),F(y))$. On suppose que l'on connaît n évaluations de f , $f(x_i)|_{i=1\dots n}$. Ces observations sont donc considérées comme des réalisations aléatoires $F(x_i)|_{i=1\dots n}$. La prédiction par

krigeage $\hat{F}(x)$ de $F(x)$ est alors définie comme la prédiction linéaire non biaisée ($\mathbb{E}(F(x)) = \mu$) qui minimise la variance de l'erreur de prédiction (appelée variance de krigeage). En d'autres termes, $\hat{F}(x)$ minimise :

$$\hat{\sigma}^2(x) = \mathbb{E} \left[(\hat{F}(x) - F(x))^2 \right] \tag{1.10}$$

parmi tous les éléments non biaisés $F(x_i)$.

On peut écrire $\hat{F}(x)$ comme une combinaison linéaire des observations :

$$\hat{F}(x) = \sum_{i=1}^n \lambda_i(x) F(x_i) \tag{1.11}$$

Le problème consiste à déterminer la pondération, i.e. les λ_i , de chacun des points environnants. Le krigeage choisit les poids à partir du degré de similarité entre les valeurs de F , *i.e.* à partir de la covariance entre les points en fonction de la distance entre ces points. Cette méthode nécessite que la moyenne et la variance de la fonction F soient stationnaires, c'est-à-dire qu'elles ne dépendent pas de la position des points mais seulement de la distance entre les points. Dans ce cas, la fonction de covariance de F sera notée $k(h) = k(\|x - y\|) = \text{Cov}(F(x), F(x + h))$. Le choix de la fonction de covariance k est une question centrale dans le cadre de l'utilisation du krigeage car elle a une influence importante sur la qualité de la prédiction. Dans [Ginsbourger 2009], les auteurs proposent des développements récents du krigeage ainsi que des conseils dans le choix du modèle à utiliser.

Au final, un krigeage donne une estimation de la forme moyenne de la surface de réponse et de l'erreur associée au modèle en chaque point, sachant que l'erreur est nulle aux points de mesure. Ce principe est illustré par la figure 1.16.

Mélange d'experts

Ce dernier type de modèle statistique n'en est pas un en soi : le principe du mélange d'experts est de mêler différents modèles statistiques, en les pondérant en fonction du degré de confiance dans chaque modèle [Jacobs 1991, Torres 2003].

Le mélange d'experts est utile pour s'affranchir du choix du type de modèle à utiliser. Le fait de construire différents modèles et d'utiliser le mélange d'experts nous permet d'obtenir le meilleur indice de confiance.

1.3.1.3 Optimisation

Reprenons la formulation classique d'un problème d'optimisation (1.1) définie dans la partie 1.3 (p. 16) :

$$\left\{ \begin{array}{l} \min_{x \in \mathcal{G}_{\text{ad}}} J_m(x) \\ \text{s.t.} \quad \left| \begin{array}{l} g_j(x) \leq 0, \quad j = 1, \dots, n_i \\ h_l(x) = 0, \quad l = 1, \dots, n_e \end{array} \right. \end{array} \right. \tag{1.12}$$

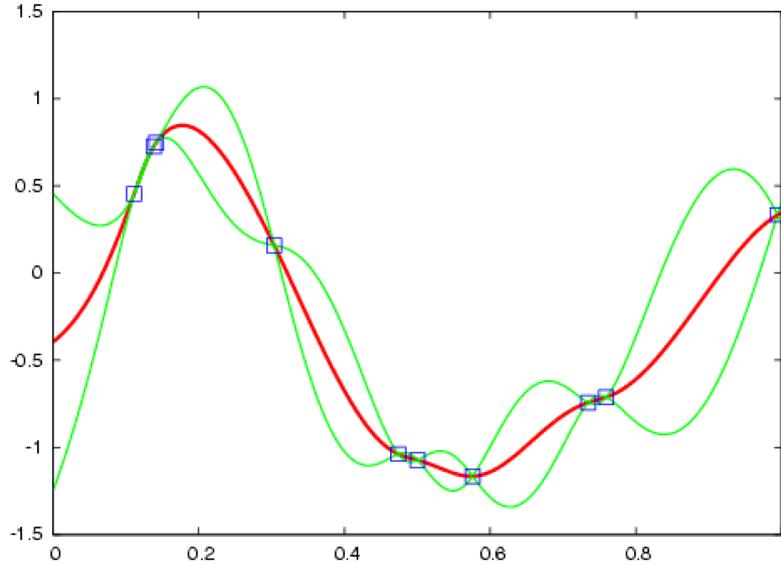


FIGURE 1.16 – Exemple de prédiction par krigeage en dimension 1 : les résultats des évaluations (carrés) sont interpolés par la prédiction \hat{f} . Les courbes vertes correspondent aux extrémités des intervalles de confiance à 95%.

Dans le cadre d'une optimisation mono-objectif, *i.e.* si m vaut 1, il s'agit de trouver le minimum de l'unique fonction coût. La recherche du minimum global peut s'avérer être une boîte de Pandore à cause de la présence de minima locaux. Il faudra s'assurer autant que possible que les algorithmes d'optimisation ne tombent pas dans ces minima locaux.

D'autre part, les problèmes traités dans le monde industriel sont souvent multi-objectifs ($m \geq 2$). Il serait alors naturel de faire un parallèle avec le cas mono-objectif et de définir une solution d'un problème multi-objectif comme une solution minimisant chacun des objectifs simultanément. Néanmoins, cette définition de l'optimalité n'a pas de sens car les objectifs sont bien souvent antagonistes. Une solution consiste à introduire la notion de dominance pour déterminer si une solution est optimale ou non. Soit x_1 et x_2 deux configurations distinctes de \mathcal{G}_{ad} , on a alors :

- $x_1 \preceq x_2$ (x_2 domine faiblement x_1) si et seulement si $\forall k \in \llbracket 1, m \rrbracket, J_k(x_2) \leq J_k(x_1)$;
- $x_1 \prec x_2$ (x_2 domine x_1) si et seulement si $x_1 \preceq x_2$ et $\exists i \in \llbracket 1, m \rrbracket$ tel que $J_i(x_2) < J_i(x_1)$.

Ceci nous permet de définir la notion d'optimalité au sens de Pareto. Elle tire son nom de son inventeur Vilfredo Pareto. La configuration x_1 est dite *Pareto optimale* si et seulement si $\nexists x_2 \in \mathcal{G}_{\text{ad}}$ tel que $x_1 \prec x_2$. Si x_1 est *Pareto optimale*, $J(x_1)$ est dit *Pareto efficace*. L'ensemble des points Pareto efficaces constitue le front de Pareto^e. Un exemple graphique de front de Pareto pour un problème bi-objectif est donné par la figure 1.17. Le front de Pareto est un puissant outil d'aide à la décision qui permet au concepteur de choisir un compromis parmi les solutions du problème d'optimisation. Pour approfondir le sujet, le lecteur pourra consulter la synthèse [Binois 2013].

e. Le front de Pareto est également appelé *surface des compromis optimaux*.

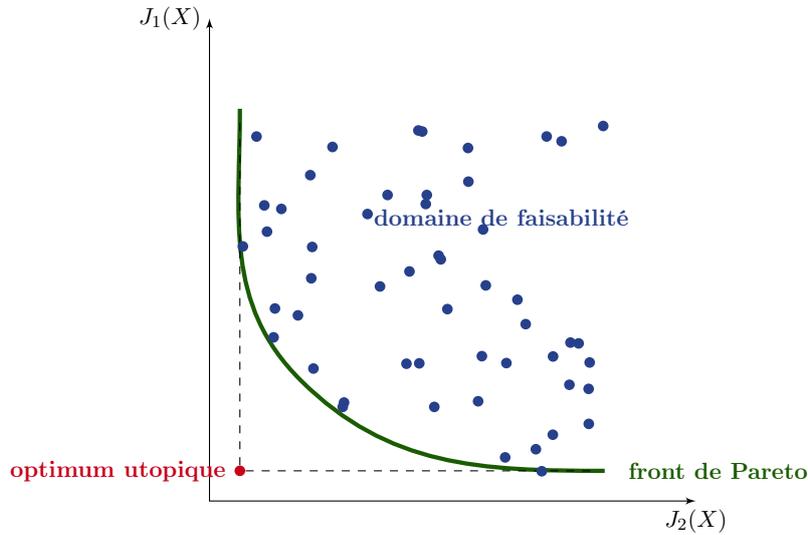


FIGURE 1.17 – Illustration du front de Pareto pour un problème bi-objectif

1.3.2 Méthode des dérivées d'ordre supérieur

L'idée est de s'appuyer sur un développement de Taylor pour construire une approximation de la fonction. Les équations de Navier-Stokes sont dérivées analytiquement et la série de Taylor de la solution est calculée à partir des dérivées d'ordre élevé. Cela permet d'obtenir une solution explicite continue en fonction des paramètres. Toutes les informations nécessaires utilisées pour déterminer la relation entre les paramètres et les variables aérodynamiques sont alors générées en une seule fois pendant le calcul et stockées dans une base de données numériques.

1.3.2.1 Principe

Cette démarche de paramétrisation est basée sur la différenciation des équations de Navier-Stokes (RANS) et sur le développement en série de Taylor à un ordre élevé de la solution [Groncin 2005]. Elle débute avec les équations stationnaires discrétisées de Navier-Stokes écrites sous la forme symbolique :

$$F(x, W(x)) = 0 \tag{1.13}$$

où :

- F représente les équations discrétisées de conservation de la masse, de la quantité de mouvement et de l'énergie ;
- $W = (\rho, \rho V, \rho E, \rho k, \rho \omega)^t$ est le vecteur des variables conservatives et de turbulence, avec :
 - ρ : la masse volumique ;
 - V : la vitesse ;
 - E : l'énergie totale ;
 - k : l'énergie cinétique de turbulence ;
 - ω : la vorticit e.
- x est le vecteur des paramètres de design.

D'après l'équation (1.13), la dérivée au premier ordre par rapport à x donne :

$$\frac{dF}{dx}(x, W) = \frac{\partial F}{\partial x}(x, W) + \frac{\partial F}{\partial W}(x, W) \cdot \frac{\partial W}{\partial x} = 0 \quad (1.14)$$

d'où :

$$(1.14) \stackrel{\Rightarrow}{(1.13)} \frac{\partial F}{\partial W}(x, W) \cdot W^{(1)} = -\frac{\partial F}{\partial x}(x, W) \quad (1.15)$$

avec $W^{(1)} = \frac{\partial W}{\partial x}$ la dérivée première de W . On note $G = \frac{\partial F}{\partial W}$ la matrice jacobienne et $R(x, W)$ le membre de droite de l'équation (1.15). Les dérivées d'ordre supérieur $W^{(n)}$ sont obtenues de manière récursive par dérivation successive de l'équation (1.15) :

$$G \cdot W^{(1)} = R \quad (1.16)$$

$$G \cdot W^{(2)} = R^{(1)} - G^{(1)} \cdot W^{(1)} \quad (1.17)$$

...

$$G \cdot W^{(n)} = R^{(n-1)} - \sum_{i=1}^{n-1} C_{n-1}^i \cdot G^{(i)} \cdot W^{(n-i)} \quad (1.18)$$

Enfin, connaissant toutes les dérivées $W^{(k)}$, on peut obtenir le nouveau champ aérodynamique correspondant aux valeurs modifiées des paramètres $x + \Delta x$ en reconstituant la série de Taylor :

$$W(x + \Delta x) = W(x) + W^{(1)} \cdot \Delta x + \dots + \frac{W^{(n)}}{n!} \cdot \Delta x^n + \mathcal{O}(\Delta x^{n+1}) \quad (1.19)$$

Ainsi, si l'on stocke tous les $W^{(k)}$, il est possible par la suite de les relire et de déterminer la valeur d'un champ aérodynamique par application de la série de Taylor pour la valeur du paramètre recherché.

1.3.2.2 Avantages et inconvénients

Une fois que l'on a calculé les termes de la série de Taylor, on a un modèle analytique de la fonction. L'évaluation d'une nouvelle solution est instantanée, la recherche d'un optimum peut donc aller vite.

En pratique, cette solution implique quelques inconvénients. D'une part, un des gros points négatifs est qu'il est nécessaire d'avoir accès au code source du solveur pour implémenter une telle méthode. D'autre part, une solution n'est pas valable sur tout le domaine mais localement autour du point nominal car on s'arrête généralement à l'ordre 2 voire 1. Qui plus est, les dérivées ne sont pas calculées analytiquement mais par différenciation automatique à chaque itération pour chaque paramètre [Aubert 2010]. Par exemple, à l'ordre 1, il faut donc résoudre x fois l'équation (1.16). À l'ordre 2, il faut résoudre l'équation (1.17) x^2 fois, et si l'on voulait aller à l'ordre 3, il faudrait résoudre l'équation correspondante x^3 fois, etc. Par conséquent, ceci limite en pratique le nombre de paramètres x à une petite valeur ($\simeq 5$). De même, le nombre de nœuds du maillage est limitant : si l'on prend un champ $u \in \mathbb{R}^n$ avec n le nombre de nœuds, sa dérivée

première par rapport aux paramètres $\partial_x u$ appartient à $\mathbb{R}^{n \times x}$, sa dérivée seconde $\partial_{xx} u$ à $\mathbb{R}^{n \times x \times x}$. En pratique, le problème est limité à l'ordre 2 et seuls les termes diagonaux du hessien sont pris, ce qui réduit encore le domaine de validité de la solution. Cette méthode peut avoir un sens pour un modèle embarqué (contrôle optimal, automatisme...) parce que les évaluations se font autour d'un même nominal. En revanche, dans le cadre d'une optimisation, on va nécessairement sortir du domaine de validité du modèle construit, auquel cas il faut réinvestir dans la construction d'un nouveau modèle. Enfin, lors de la modification d'un paramètre géométrique, le maillage est morphé. La qualité du morphing est donc tributaire des méthodes disponibles dans le logiciel et qui plus est, le morphing a ses propres limitations, en particulier en mécanique des fluides (déformation de la couche limite).

1.3.3 Optimisation directe (*gradient-based*)

1.3.3.1 Principe

Les méthodes d'optimisation par gradient nécessitent une bonne connaissance de la fonction étudiée. En effet, il faut être capable de calculer le gradient, ce qui implique d'une part que la fonction soit suffisamment régulière et d'autre part d'être intrusif dans les codes de calcul. Parmi les algorithmes de type gradient, il faut distinguer les cas avec ou sans contraintes (cf. [Fletcher 1980, Culioli 1994, Boyd 2004, Nocedal 2006, Allaire 2007] pour un état de l'art complet sur le sujet).

L'algorithme de gradient repose sur le fait que la fonction J décroît, dans le voisinage d'un point x , le plus fortement dans la direction opposée à celle du gradient de J en x , à savoir dans la direction $-\nabla J(x)$. De proche en proche, l'algorithme va converger vers un minimum local, voire global si la fonction est strictement convexe (*i.e.* si elle est toujours inférieure à son approximation linéaire). Ci-dessous est présenté un exemple d'algorithme sans contraintes.

Algorithme du gradient – Soit J une fonction réelle différentiable : $x \in \mathbb{R}^n \mapsto J(x) \in \mathbb{R}$ et $\nabla J(x) = \left(\frac{\partial J}{\partial x_1}, \dots, \frac{\partial J}{\partial x_n} \right)^t$ son gradient. On se donne un point initial $x^0 \in \mathbb{R}$ et un seuil de tolérance $\varepsilon \geq 0$. L'algorithme du gradient consiste à « se déplacer » d'une itérée en suivant la ligne de plus grande pente :

1. calcul de $J(x^k)$ et de $\nabla J(x^k)$;
2. test d'arrêt : fin de l'algorithme si $\|\nabla J(x^k)\| \leq \varepsilon$;
3. calcul du pas $\beta^k > 0$;
4. nouvelle itérée : $x^{k+1} = x^k - \beta^k \nabla J(x^k)$.

On parle d'algorithme de *gradient à pas optimal*^f lorsque l'on cherche β^k par une recherche linéaire sur J en x^k le long de la direction $-\nabla J(x^k)$. On prend alors $\beta^k = \arg \min_{\beta > 0} J(x^k - \beta \nabla J(x^k))$.

Il est possible aussi de prendre β^k constant, on parle alors d'*algorithme à pas constant*. Dans le même cadre (algorithmes sans contraintes), on peut également citer les *méthodes de Newton, du gradient conjugué, de Quasi-Newton...*

f. Cet algorithme est également appelé *algorithme de la plus grande pente*.

1.3.3.2 Méthode adjointe en mécanique des fluides

En mécanique des fluides, le calcul du gradient fait souvent écho à l'approche adjointe. Cette approche vise à évaluer la sensibilité d'une prestation J (perte de charge, coefficient de traînée aérodynamique...). Elle est couramment utilisée dans le monde de l'aéronautique depuis les années 1990, mais est arrivée tardivement dans le secteur automobile [Othmer 2005]. Plusieurs méthodes permettent de résoudre ce problème, en utilisant par exemple :

- les différences finies ;
- l'approche adjointe discrète ;
- l'approche adjointe continue.

Historiquement, la méthode adjointe en mécanique des fluides a été introduite par O. Pironneau [Pironneau 1984] et appliquée aux écoulements régis par des EDP elliptiques. A. Jameson a été le premier à présenter la formulation de l'adjoint continu pour la conception optimale de formes aérodynamiques pour des écoulements transsoniques non visqueux [Jameson 1988, Jameson 1995, Jameson 1994]. Par la suite, [Jameson 1998] propose une extension aux écoulements visqueux où les équations de Navier-Stokes sont utilisées comme équations d'état. Un état de l'art complet sur la méthode adjointe est donné dans [Thévenin 2008].

1.3.3.3 Calcul des gradients par différences finies

Cette méthode a été utilisée jusque dans les années 1990. Elle a tendance à disparaître au bénéfice des deux méthodes adjointes présentées par la suite. En effet, les problèmes en mécanique des fluides impliquent généralement un grand nombre de paramètres de forme x_i . Or, cette démarche nécessite – en plus de l'évaluation de la solution courante $J(x)$ – autant de calculs qu'il y a de paramètres x_i pour une évaluation à l'ordre un des dérivées $\frac{\partial J}{\partial x_i}$, soit la résolution de $n + 1$ problèmes non linéaires. Autant dire que cela est souvent rédhibitoire.

$$\frac{\partial J}{\partial x_i} \simeq \frac{J(x_i + \delta x_i) - J(x_i)}{\delta x_i} \tag{1.20}$$

L'approche adjointe qui suit est plus efficace pour calculer le gradient de la fonction objectif et ne nécessite que la résolution de systèmes linéaires. Le coût total pour le calcul du gradient ne dépend pas du nombre de variables et est approximativement inférieur à celui de l'évaluation de la solution courante.

1.3.3.4 Calcul des gradients par l'équation adjointe

Il existe plusieurs formalismes pour arriver aux équations adjointes. Nous présentons ici la plus classique qui consiste à introduire un lagrangien associé à la fonction objectif J . On appelle F les équations d'état (équations d'Euler, de Navier-Stokes...) :

$$F(x, W(x)) = 0 \tag{1.21}$$

avec x les variables d'optimisation (position des nœuds, porosité des cellules...) et W le vecteur d'état des variables conservatives. Le gradient de ces contraintes est donné par :

$$\frac{dF}{dx} = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial W} \cdot \frac{\partial W}{\partial x} = 0 \tag{1.22}$$

La fonction lagrangienne est donnée par :

$$\mathcal{L}(x, W(x)) = J(x, W(x)) + \lambda^t F(x, W(x)) \tag{1.23}$$

où λ représente le multiplicateur de Lagrange, ainsi que le vecteur adjoint. L'équation adjointe du problème de calcul de gradient est obtenue en imposant $\frac{d\mathcal{L}}{dx} = 0$, *i.e.* :

$$\frac{d\mathcal{L}}{dx} = \left(\frac{\partial J}{\partial x} + \lambda^t \frac{\partial F}{\partial x} \right) + \left(\frac{\partial J}{\partial W} + \lambda^t \frac{\partial F}{\partial W} \right) \frac{\partial W}{\partial x} = 0 \tag{1.24}$$

et en choisissant λ de façon à s'affranchir du calcul du terme $\frac{\partial W}{\partial x}$:

$$\frac{\partial J}{\partial W} + \lambda^t \frac{\partial F}{\partial W} = 0 \tag{1.25}$$

ce qui s'écrit encore (et que l'on appelle *équation adjointe* à $F(x, W(x))$) :

$$\left[\frac{\partial F}{\partial W} \right]^t \lambda = - \left[\frac{\partial J}{\partial W} \right]^t \tag{1.26}$$

In fine, le gradient de la fonction objectif J par rapport aux variables d'optimisation x est égal à l'expression finale du Lagrangien :

$$\begin{aligned} \frac{dJ}{dx} &= \frac{\partial J}{\partial x} + \frac{\partial J}{\partial W} \frac{\partial W}{\partial x} \stackrel{(1.25)}{\Rightarrow} \frac{dJ}{dx} = \frac{\partial J}{\partial x} - \lambda^t \frac{\partial F}{\partial W} \frac{\partial W}{\partial x} \\ &\stackrel{(1.22)}{\Rightarrow} \frac{dJ}{dx} = \frac{\partial J}{\partial x} + \lambda^t \frac{\partial F}{\partial x} = \frac{d\mathcal{L}}{dx} \end{aligned} \tag{1.27}$$

Si l'on regarde plus en détail l'équation (1.27), on a :

$$\frac{dJ}{dx} = \underbrace{\frac{\partial J}{\partial x}}_{\text{dérivation relative à la géométrie}} + \underbrace{\lambda^t \frac{\partial F}{\partial x}}_{\text{dérivation relative à la physique}}$$

En prenant l'exemple d'un profil d'aile d'avion, on a :

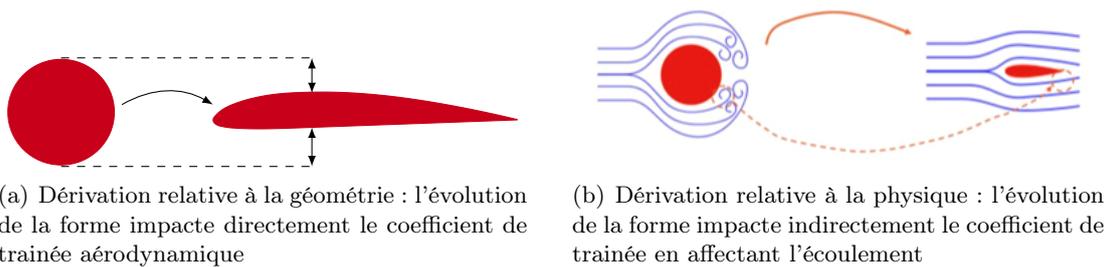


FIGURE 1.18 – Influence de chaque dérivée

L'équation (1.26) est résolue par un *solveur adjoint*. Il existe deux approches : une dite *continue* et une dite *discrète*. La différence qui permet de distinguer ces deux approches réside dans la manière dont les équations adjointes discrètes sont obtenues à partir des équations d'états. La figure 1.19 présente schématiquement les deux façons alternatives d'obtenir les équations adjointes discrètes.

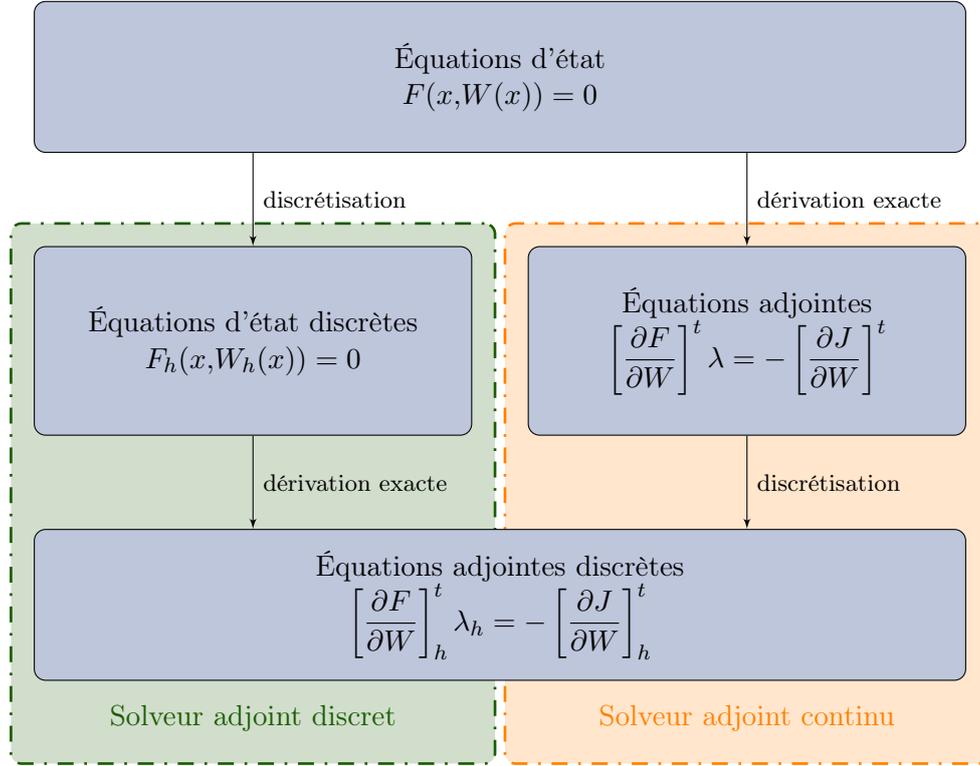


FIGURE 1.19 – Schéma des deux approches pour parvenir aux équations adjointes discrètes

Comparaison des méthodes discrète et continue

Pour le cas discret, la dérivation des équations d'état discrètes est plus lourde, même lorsqu'une différenciation automatique est utilisée. Une fois que les équations ont été trouvées, elles sont résolues en utilisant le même solveur que celui qui est utilisé pour résoudre les équations d'état. Cette démarche est dépendante du solveur mais elle est plus précise car les gradients sont cohérents avec le problème discret. Toutefois, il y a un risque qu'il n'y ait plus de cohérence entre les équations adjointes « exactes » $\left[\frac{\partial F}{\partial W} \right]^t \lambda = - \left[\frac{\partial J}{\partial W} \right]^t$ et les équations adjointes discrètes $\left[\frac{\partial F}{\partial W} \right]_h^t \lambda_h = - \left[\frac{\partial J}{\partial W} \right]_h^t$.

En utilisant l'approche continue, on peut utiliser différents schémas de discrétisation. Les équations adjointes peuvent par conséquent être dérivées et résolues sans avoir accès au code source du solveur primaire. Cependant, dans la pratique, il a été prouvé que la précision de l'approche adjointe continue dépend du schéma de discrétisation qui doit être aussi proche que possible de celui utilisé pour la discrétisation des équations d'état [Anderson 1999]. En outre, à chaque fois que l'on souhaite ajouter un nouveau modèle physique (modèle de turbulence, etc.), il est nécessaire de dériver les équations adjointes continues pour ce problème et de les discrétiser. Enfin, pour certaines conditions aux limites, la définition des équations adjointes continues est délicate voire problématique [Peter 2006].

Des différences dans les valeurs des gradients, calculés par les deux approches, peuvent apparaître aussi si le maillage n'est pas assez fin.

Solveurs adjoints

De nombreux solveurs adjoints (FLUENT[®] (ANSYS), Star-CCM+[®] (CD-Adapco), Pamflow[®] (ESI), OpenFOAM[®] (ESI)...) sont disponibles actuellement sur le marché, la plupart étant développés sur une approche discrète. Ils permettent d'évaluer entre autres les sensibilités des critères suivants :

- la perte de charge ;
- l'uniformité de l'écoulement, en définissant par exemple l'index d'uniformité γ pour une section A comme :

$$\gamma = 1 - \frac{1}{2A\bar{v}_n} \int_{outlet} |v_n - \bar{v}_n| dA \quad (1.28)$$

avec \bar{v}_n la vitesse normale moyenne dans la section A . Une valeur de $\gamma = 1$ correspond à un écoulement totalement uniforme ;

- une force ;
- un moment.

Utilisation des gradients

Le calcul des adjoints peut être par la suite utilisé dans le cadre d'une optimisation topologique. Cette méthode est arrivée tardivement dans le domaine de la mécanique des fluides, notamment après [Anderson 2003]. Le fonctionnement est comparable à ce qui se fait déjà en mécanique des structures, sauf que l'on cherche une répartition de perméabilité optimale représentée par un terme de Darcy. La vertu d'une optimisation topologique réside plus dans le fait de trouver des designs novateurs que dans la mise au point fine de la géométrie. Il est par conséquent intéressant d'utiliser ensuite une optimisation géométrique se basant sur les gradients.

La géométrie est souvent déformée en utilisant un morphing, basé par exemple sur un opérateur de lissage de type laplacien. Ceci est problématique en mécanique des fluides notamment à cause des couches limites utilisées pour capter les turbulences au niveau des parois. En effet, au bout de quelques itérations, la couche limite est totalement déformée, ce qui peut fortement perturber les résultats.

1.4 Synthèse

Nous avons abordé dans ce chapitre les différentes familles de paramétrage dédiées à la conception de pièces. Nous souhaitons optimiser des pièces automobiles en utilisant les paramètres métier et en intégrant les contraintes de fabrication. Pour ces raisons, nous nous concentrons dans la suite du manuscrit sur des optimisations basées sur des modèles CAO. On écarte les optimisations topologiques et géométriques car elles requièrent notamment de savoir mathématiser toutes les contraintes du cahier des charges. De même, le morphing ne permet pas d'accéder aux paramètres métier et ne parvient pas à préserver la qualité de maillage.

Dans un second temps, nous avons présenté les méthodes d'optimisation (*gradient-free* ou *gradient-based*) qui sont utilisées au fil du manuscrit. Le chapitre qui suit utilise une méthode d'optimisation par plans d'expériences et surfaces de réponse. Cette approche ne permet que de traiter un nombre limité de paramètres. Or, comme les modèles CAO peuvent compter jusqu'à

plusieurs centaines de paramètres, nous traitons de la réduction du nombre de paramètres *a priori*, c'est-à-dire avant de réaliser les simulations numériques. Dans un second temps, on s'intéressera à limiter le coût d'un calcul qui peut durer plusieurs jours (sur 64 cœurs) en mécanique des fluides.

Bibliographie du 1^{er} chapitre

- [Allaire 2003] G. Allaire, F. Jouve et A.M. Toader. *Structural optimization using sensitivity analysis and a level-set method*, 2003. (cité page 17)
- [Allaire 2004] G. Allaire et F. Jouve. *A level-set method for vibration and multiple loads structural optimization*, 2004. (cité page 17)
- [Allaire 2007] G. Allaire. *Conception optimale de structures*. Springer Berlin Heidelberg, 2007. (cité pages 16 et 26)
- [Anderson 1999] W.K. Anderson et V. Venkatakrishnan. *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation*. *Computers and Fluids*, vol. 28, page 443–480, 1999. (cité page 29)
- [Anderson 2003] W.K. Anderson et V. Venkatakrishnan. *Topology optimization of fluids in Stokes flow*. *Int. J. Num. Meth. Fluids*, vol. 41, page 77, 2003. (cité page 30)
- [Aubert 2010] S. Aubert, P. Ferrand, F. Pacull et M. Buisson. *Fast CFD for shape and flow parameterization with meta-models built on high-order derivatives - Applications to fast design*. 27th International Congress of the Aeronautical Science, 2010. (cité page 25)
- [Bendsoe 2003] M.P. Bendsoe et O. Sigmund. *Topology optimization : theory, methods and applications*. Springer Berlin Heidelberg, 2003. (cité page 12)
- [Benoist 1994] D. Benoist, Y. Tourbier et S. Germain-Tourbier. *Plans d’expériences : construction et analyse*. Technique & Documentation-Lavoisier, 1994. (cité pages 18, 19 et 129)
- [Binois 2013] M. Binois. *Synthèse sur l’optimisation multi-objectifs*. Rapport technique, Consortium ReDICE, 2013. (cité page 23)
- [Boyd 2004] S. Boyd et L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. (cité page 26)
- [Brujic 2010] D. Brujic, M. Ristic, M. Mattone, P. Maggiore et G.P. De Poli. *CAD based shape optimization for gas turbine component design*. *Structural and Multidisciplinary Optimization*, vol. 41, pages 647–659, 2010. (cité page 10)
- [Cottrell 2009] J.A. Cottrell, T.J.R. Hughes et Y. Bazilevs. *Isogeometric analysis, toward integration of CAD and FEA*. John Wiley and Sons, 2009. (cité page 12)
- [Culioli 1994] J.C. Culioli. *Introduction à l’optimisation*. Ellipses, 1994. (cité pages 26 et 115)
- [Dapogny 2013] C. Dapogny. *Shape optimization, level set methods on unstructured meshes and mesh evolution*. PhD thesis, UPMC, 2013. (cité pages 13 et 14)
- [de Falco 2011] C. de Falco, A. Reali et R. Vázquez. *GeoPDEs : a research tool for isogeometric analysis of PDEs*. *Advances in Engineering Software*, vol. 42, no. 12, pages 1020–1034, 2011. (cité page 12)
- [Devalan 2009] P. Devalan. *Simulation numérique dans le processus de conception de systèmes mécaniques*. *Techniques de l’ingénieur*, no. bm5013, 2009. (cité page 5)
- [du Cauzé de Nazelle 2013] P. du Cauzé de Nazelle. *Paramétrage de formes surfaciques pour l’optimisation*. PhD thesis, École Centrale de Lyon, 2013. (cité page 12)
- [Dupuy 2007] D. Dupuy et C. Helbert. *Retour d’expériences sur métamodèles : partie théorique*. Rapport technique, Consortium DICE, 2007. (cité page 21)

- [Fletcher 1980] R. Fletcher. Practical methods of optimization. John Wiley & Sons, 1980. (cité page 26)
- [Friedman 1991] J.H. Friedman. *Multivariate Adaptive Regression Splines*. The annale of statistics, vol. 19, pages 1–141, 1991. (cité page 21)
- [Ginsbourger 2009] D. Ginsbourger, D. Dupuy, A. Badea, L. Carraro et O. Roustant. *A note on the choice and the estimation of kriging models for the analysis of deterministic computer experiments*. Applied Stochastic Models in Business and Industry, vol. 25, pages 115–131, 2009. (cité page 22)
- [Goupy 1999] J. Goupy. Plans d’expériences pour surfaces de réponse. Dunod, 1999. (cité pages 19 et 130)
- [Grondin 2005] G. Grondin, V. Kelner, P. Ferrand et S. Moreau. *Robust design and parametric performance study of an automotive fan blade by coupling multi-objective genetic optimization and flow parameterization*. Proceedings of the International Congress of Fluid Dynamics Applications in Ground Transportation, 2005. (cité page 24)
- [Haskins 2007] C. Haskins. Systems engineering handbook : a guide for system life cycle processes and activities. Numéro INCOSE-TP-2003-002-03.1. International Council on Systems Engineering, 2007. (cité page 4)
- [Hughes 2005] T.J.R. Hughes, J.A. Cottrell et Y. Bazilevs. *Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement*. Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 39-41, pages 4135–4195, 2005. (cité pages 11 et 12)
- [Jacobs 1991] R.A. Jacobs, M.I. Jordan, S.J. Nowlan et G.E. Hinton. Adaptive mixture of local experts, volume 3. 1991. (cité pages 20 et 22)
- [Jameson 1988] A. Jameson. *Aerodynamic design via control theory*. Journal of Scientific Computing, vol. 3, pages 233–260, 1988. (cité page 27)
- [Jameson 1994] A. Jameson et J. Reuther. *Control theory based airfoil design using the Euler equations*. AIAA, 1994. (cité page 27)
- [Jameson 1995] A. Jameson. *Optimum aerodynamic design using CFD and control theory*. AIAA, 1995. (cité page 27)
- [Jameson 1998] A. Jameson, N. Pierce et L. Martinelli. *Optimum aerodynamic design using the Navier-Stokes equations*. AIAA, 1998. (cité page 27)
- [Kooperberg 1997] C. Kooperberg, S. Bose et C.J. Stone. *Polychotomous regression*. Journal of the American statistical association, vol. 92, pages 117–127, 1997. (cité page 21)
- [Lebrun 2003] M. Lebrun. *Simulation et CAO en automatique et mécatronique*. Techniques de l’ingénieur, no. s7260, 2003. (cité page 4)
- [Luzeaux 2011] D. Luzeaux, J.R. Ruault et J.L. Wippler. Maîtrise de l’ingénierie des systèmes complexes et des systèmes de systèmes : étude de cas. Hermès science publications-Lavoisier, 2011. (cité page 4)
- [Matheron 1963] G. Matheron. *Traité de géostatistique appliquée, 2 : le krigeage*. Technip, 1963. (cité page 21)
- [Michailidis 2014] G. Michailidis. *Manufacturing constraints and multi-phase shape and topology optimization via a level-set method*. PhD thesis, École Polytechnique, 2014. (cité page 13)

- [Moos 2004] O. Moos, F. Klimetzek et R. Rossmann. *Bionic optimization of air-guiding systems*. SAE Technical Paper, 2004. (cité pages 13 et 15)
- [Nocedal 2006] J. Nocedal et S. Wright. Numerical optimization. John Wiley & Sons, 2006. (cité page 26)
- [Othmer 2005] C. Othmer et Th. Grahs. *Approaches to fluid dynamic optimization in the car development process*. EUROGEN, 2005. (cité page 27)
- [Peter 2006] J. Peter, S. Burguburu et M. Marcelet. Introduction à l'optimisation de forme en aérodynamique et quelques exemples d'application. ONERA, 2006. (cité page 29)
- [Pironneau 1984] O. Pironneau. Optimal shape design for elliptic systems. Springer-Verlag, 1984. (cité page 27)
- [Roy 2008] R. Roy, S. Hinduja et R. Teti. *Recent advances in engineering design optimisation : challenges and future trends*. CIRP Annals - Manufacturing Technology, vol. 57, pages 697–715, 2008. (cité pages 8 et 10)
- [Saitou 2005] K. Saitou, K. Izui, S. Nishiwaki et P. Papalambros. *A survey of structural optimization in mechanical product development*. Journal of Computing and Information Science in Engineering, vol. 5, pages 214–226, 2005. (cité pages 8 et 10)
- [Staten 2011] M.L. Staten, S.J. Owen, S.M. Shontz, A.G. Salinger et T.S. Coffey. *A comparison of mesh morphing methods for 3D shape optimization*. Proceedings of the 20th international meshing round-table, pages 293–311, 2011. (cité page 11)
- [Suh 2002] M.W. Suh, J.H. Lee, K.Y. Cho et S.I. Kim. *Section property method and section shape method for the optimum design of vehicle body structures*. International Journal of Vehicle Design, vol. 30, pages 115–134, 2002. (cité page 9)
- [Thévenin 2008] D. Thévenin et G. Janiga. Optimization and computational fluid dynamics. Springer, 2008. (cité page 27)
- [Torres 2003] R. Torres, R. Salas, H. Allende et C. Moraga. *Robust expectation maximization learning algorithm for mixture of experts*. Computational Methods in Neural Modeling, vol. 2686, pages 238–245, 2003. (cité page 22)
- [Yang 1992] R.J. Yang, A. Lee et D.T. McGeen. *Application of basis function concept to practical shape optimization problems*. Structural optimization, vol. 5, pages 55–63, 1992. (cité page 11)

Réduction de la dimension du problème et multiniveaux de fidélité

Sommaire

2.1	Construction d'un modèle paramétré	36
2.1.1	Contexte	36
2.1.2	Reconstruction surfacique	38
2.1.2.1	Nettoyage des surfaces	38
2.1.2.2	Extraction et discrétisation des courbes du contour	38
2.1.2.3	Discrétisation de la surface	38
2.1.2.4	Création du réseau de courbes	41
2.1.2.5	Création de la surface sur réseau	43
2.1.3	Paramétrisation	44
2.1.4	Application à un cas industriel	46
2.2	Analyse hors simulation d'un modèle paramétré	46
2.2.1	Détection des zones interdites	46
2.2.2	Gestion des zones interdites par machine à vecteurs de support	51
2.2.2.1	Formulation mathématique pour des données séparables	52
2.2.2.2	Formulation mathématique pour des données non séparables	54
2.2.2.3	Transformations non linéaires	55
2.2.3	Gestion des zones interdites par génération d'un grand nombre d'essais	57
2.3	Réduction du nombre de paramètres <i>a priori</i>	58
2.3.1	1 ^{re} voie : voxellisation et analyse en composantes principales	58
2.3.1.1	Voxellisation	59
2.3.1.2	Analyse en composantes principales	59
2.3.1.3	Exemple d'application	61
2.3.1.4	Autres techniques	66
2.3.2	2 ^e voie : réseau de neurones artificiels	67
2.4	Approche multi-niveaux de fidélité	69
2.5	Application de la méthode à un cas industriel	70
2.5.1	Description du cas industriel	71
2.5.2	Construction du modèle CAO	72
2.5.2.1	Véhicule	72
2.5.2.2	Veine de la soufflerie	73
2.5.3	Modélisation de la frontière du domaine de conception	74
2.5.4	Création des méta-paramètres	75
2.5.5	Génération du plan d'expériences	76

2.5.6	Simulations	77
2.5.6.1	Création des fichiers	77
2.5.6.2	Maillage	77
2.5.6.3	Modèle physique	78
2.5.7	Gestion de la boucle de calcul	78
2.5.8	Résultats	79
2.6	Discussion de l'application	80
	Bibliographie du 2 ^e chapitre	83

Dans ce chapitre, on va se concentrer sur une méthode d'optimisation utilisant des plans d'expériences. La boucle reposera sur un modèle CAO paramétré à construire à partir d'une surface morte. De ce modèle seront issus des paramètres CAO dont on cherchera dans un second temps à réduire le nombre. Enfin, les simulations numériques prennent parfois plusieurs jours comme en mécanique des fluides. Nous verrons comment composer avec de tels calculs.

2.1 Construction d'un modèle paramétré

2.1.1 Contexte

La mise en place de boucles d'optimisation est quelque chose d'assez récent dans un cadre industriel. La conception des produits est encadrée depuis des années par des *règles métier* qui permettent d'assurer un niveau de performance minimal à la pièce étudiée. Ces règles métier font intervenir des paramètres métier dont on connaît l'influence. On sait par exemple que pour optimiser le coefficient de traînée aérodynamique, on pourra jouer sur les paramètres liés à la position du bord de fuite, à la resserrée des parties latérales et à la position du bas de bouclier arrière (voir la figure 2.1).



FIGURE 2.1 – Paramètres d'équilibre du sillage d'un véhicule

Les boucles d'optimisation dont on parle sont situées dans une phase avancée de la conception du produit. Ainsi, les modèles de simulation sont très prédictifs mais nous sommes en revanche très contraints (cf. figure 1.2). En effet, il est nécessaire de prendre en compte les contraintes liées aux processus de fabrication (angles de dépouille, congés, épaisseurs minimales...) et au thème design (lignes de carre du véhicule, continuités géométriques, ruptures géométriques, etc.), ce qui milite pour une optimisation de type paramétrique.

Tout concepteur aimerait disposer d'un modèle CAO avec un historique de construction. Cet historique représente les objets manipulés et les opérations qui leur ont été appliquées. Pourtant,

entre les différents départements d'une entreprise, il n'existe pas toujours de support numérique exploitable que l'on puisse échanger. Par exemple, les concepteurs et les *designers* (au sens artistique du terme) ne partagent pas les mêmes outils. Le transfert des modèles numériques se fait donc avec quelques difficultés. Les concepteurs récupèrent une surface morte^a avec laquelle ils vont travailler, sans la reconstruire.

En résumé, notre objectif est de développer une méthode qui mette à la disposition des concepteurs un support numérique exploitable. Si l'on reprend le cas précédent, ce modèle permettrait par exemple de mesurer numériquement les sensibilités des paramètres aérodynamiques. Les concepteurs bénéficieraient au passage d'un outil d'aide à la décision lors des arbitrages entre *prestations du véhicule* et *design*. Le nouveau support numérique permettrait de se substituer le plus possible aux séances en soufflerie, en diminuant le nombre de maquettes physiques, les essais coûteux et les délais de développement du véhicule par l'introduction de boucles numériques courtes.

À partir de la surface morte, plusieurs approches sont envisageables afin d'exploiter le support numérique :

- réparer la surface, puis la mailler. Il n'est pas possible dans ce cas d'utiliser des paramètres CAO. Ainsi, dans le cas où l'on souhaiterait optimiser la forme, il serait donc nécessaire d'utiliser un morphing de maillage. Si ce dernier est aisé à mettre en œuvre, notamment parce qu'il ne nécessite pas de remaillage, il rend complexe l'accès aux paramètres métier. En outre, le morphing ne permet pas de contrôler finement la déformation du maillage, c'est-à-dire qu'une perturbation du maillage fait souvent intervenir plusieurs paramètres métier. Il n'est donc pas possible de connaître précisément l'influence liée à chaque paramètre.
- reconstruire la surface. Cette méthode a le gros inconvénient de nécessiter une étape supplémentaire dans le processus numérique, ce qui engendre donc un délai pour la reconstruction de la surface, en plus de nécessiter des compétences CAO. Néanmoins elle a pour mérite de rendre accessible les paramètres métier, de maîtriser finement la déformation de la géométrie et de mettre à disposition un support opérationnel relisible par tous. Ce support est directement exploitable dans une optimisation.

Nous proposons donc une méthode qui permette de passer d'une surface morte à un modèle paramétré. Cette approche consiste dans un premier temps à reconstruire les surfaces puis à les paramétrer :

- nettoyage de la surface morte ;
- extraction des courbes de cette surface ;
- discrétisation de la surface morte ;
- création du réseau de courbes ;
- création de la surface sur réseau ;
- paramétrage de la nouvelle surface.

a. On appellera par la suite *surface morte* toute surface issue d'un environnement CAO dont l'historique de construction est inconnu.

2.1.2 Reconstruction surfacique

2.1.2.1 Nettoyage des surfaces

La géométrie issue du département *Design* n'est pas exploitable car de nombreuses surfaces sont superposées, non connexes, ou encore la continuité géométrique des éléments est inexistante. Les logiciels *CAO* disponibles sur le marché intègrent des outils permettant de remédier à ces problèmes. Notons que si l'on souhaitait mailler cette géométrie, cette étape pourrait s'avérer très longue. Dans notre cas, nous souhaitons reconstruire la géométrie à l'aide de plusieurs surfaces. La définition de la représentation d'une surface se fait à partir de surfaces élémentaires. Les raccords entre ces dernières assurent la continuité de la surface totale. Chaque surface que l'on cherche à reconstruire peut donc être l'assemblage de plusieurs patches. Le nettoyage consiste à supprimer les entités inutiles, rapiécer certains éléments et fusionner l'ensemble des patches constituant la surface.

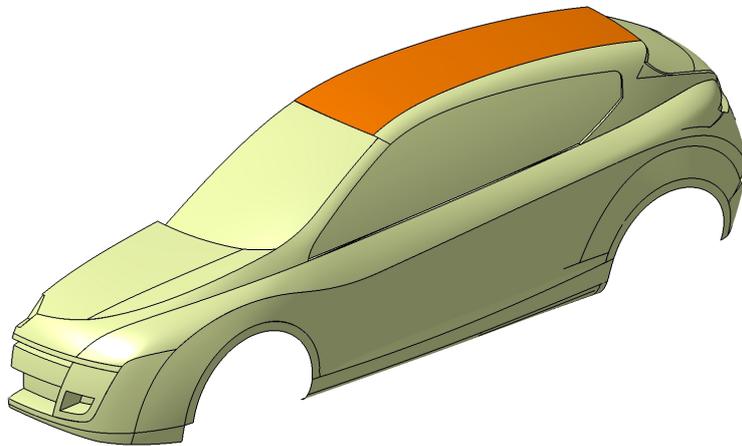


FIGURE 2.2 – Sélection du pavillon

2.1.2.2 Extraction et discrétisation des courbes du contour

Chaque courbe qui constitue le bord de la surface considérée est extraite individuellement. Si l'un des bords est formé de plusieurs éléments, ces éléments sont assemblés afin de ne constituer qu'une seule courbe. Les courbes obtenues sont ensuite discrétisées : le nombre de points de discrétisation varie en fonction de la longueur de la courbe. On distingue plusieurs types de point : les points pilotants qui vont diriger les déformations par la suite. Ce sont souvent des points remarquables (extrémité d'une courbe, point d'inflexion...). Les autres points servent juste à la qualité de reconstruction de la courbe.

2.1.2.3 Discrétisation de la surface

On veut déformer la surface mais on n'a pas accès aux équations qui la régissent. On va donc plaquer un réseau de courbes sur la surface et appliquer la déformation à ces courbes (cf. paragraphe 2.1.3 (p. 44)). Puis, on reconstruira la surface en s'appuyant sur ce réseau.

Il y a deux grosses approches pour tracer un réseau de courbes sur une surface gauche :

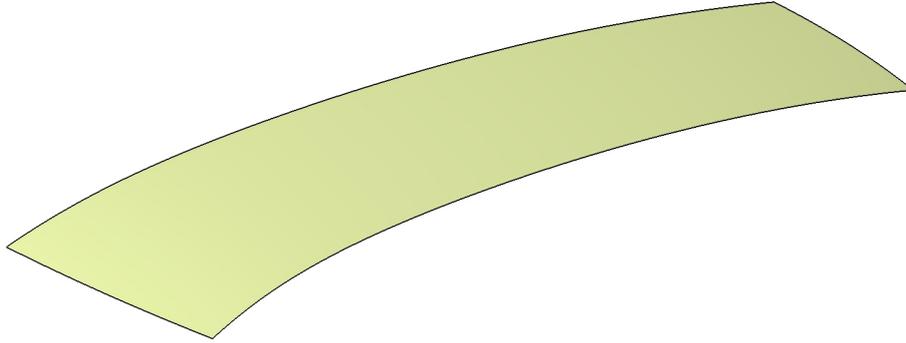


FIGURE 2.3 – Extraction de la surface à reconstruire

- la première utilise une paramétrisation harmonique. Cette technique est détaillée dans le chapitre 3 (p. 85) pour une tout autre utilisation. Elle nécessite de résoudre un système de Laplace, ce qui n'est pas aisé à mettre en œuvre en utilisant l'API d'un système CAO ;
- la seconde approche est plus heuristique. On rappelle que le bord de la surface est connu. On veut plaquer un réseau de courbes parallèles sur la surface. Pour cela, il suffit d'opérer une double interpolation. Les points ainsi obtenus ne seront pas nécessairement sur la surface mais en seront assez proches, on pourra alors les projeter. Il est à noter que les systèmes CAO peuvent fournir les lignes isoparamétriques d'une surface 3D. Cependant, ils se basent sur une estimation de la surface originale. Or, cette dernière a bien souvent été modifiée et les lignes ne correspondent à rien.

1^{re} étape : interpolation

Pour mettre en œuvre le deuxième point, nous nous inspirons des méthodes de transport-projection. Pour simplifier l'exposé, nous considérons que la surface à reconstruire Ω est suffisamment régulière (*i.e.* qu'il n'y a pas de variation soudaine) et que son contour peut être défini sous la forme de 4 côtés de telle sorte que le nombre de points sur 2 côtés logiquement connectés soit égal. On définit alors $\tilde{\Omega} \in \mathbb{R}^2$ le carré de côté unité, dont les côtés ont la même discrétisation que le contour de la surface Ω . En joignant les points qui se font face, on construit un maillage canonique de $\tilde{\Omega}$ (voir la figure 2.4).

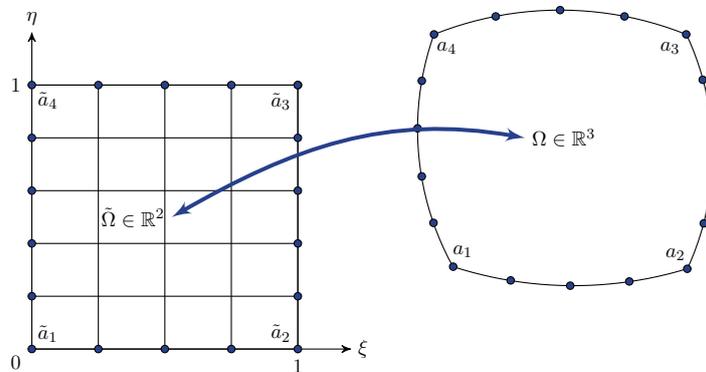


FIGURE 2.4 – Transport-projection sur le carré unité : seuls les points du bord sont connus.

Chapitre 2. Réduction de la dimension du problème et multiniveaux de fidélité

Il reste à définir une fonction F permettant de projeter tout point de $\tilde{\Omega}$ sur Ω en assurant des propriétés convenables. Soient $a_i|_{i=1\dots 4}$ les quatre sommets de ce quadrangle et $a_{i+4}|_{i=1\dots 4}$ quatre points situés chacun sur un de ses côtés. On cherche F comme une fonction classique d'interpolation des méthodes d'éléments finis :

$$F(\xi, \eta) = \sum_{i=1}^8 L_i(\xi, \eta) a_i \quad (2.1)$$

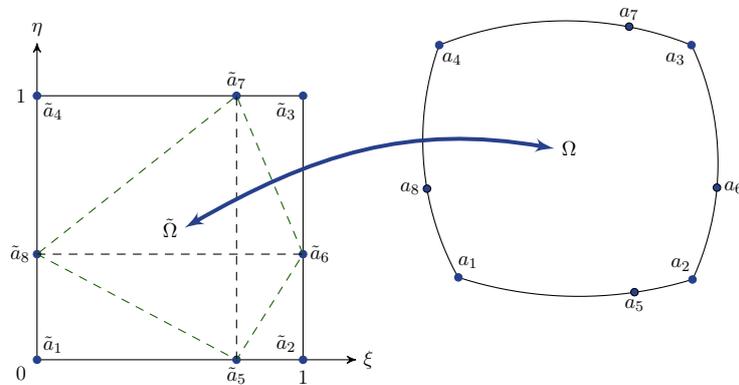


FIGURE 2.5 – Recherche des fonctions d'interpolation des méthodes d'éléments finis

On construit les fonctions L_i de façon à ce que $\forall i, j \in \llbracket 1, 8 \rrbracket$, $L_i(a_j) = \delta_{ij}$. Par exemple, pour le point \tilde{a}_3 , on cherche à annuler la fonction L_3 pour tous les points $\tilde{a}_i|_{i \neq 3}$. $\xi = 0$ annule \tilde{a}_1 , \tilde{a}_4 et \tilde{a}_8 ; $\eta = 0$ annule \tilde{a}_1 , \tilde{a}_2 et \tilde{a}_5 . Pour annuler \tilde{a}_6 et \tilde{a}_7 , on a besoin du segment passant par ces deux points. Le cas général nous donne pour le segment $[\tilde{a}_i \tilde{a}_j]$:

$$\eta = \frac{\eta_j - \eta_i}{\xi_j - \xi_i} (\xi - \xi_i) + \eta_i \quad (2.2)$$

c'est-à-dire que l'on a pour le segment $[\tilde{a}_6 \tilde{a}_7]$: $\eta = \frac{1 - \eta_6}{\xi_7 - 1} (\xi - 1) + \eta_6$, soit encore :

$$\frac{\eta_6}{1 - \eta_6} + \frac{1}{1 - \xi_7} - \frac{\xi}{1 - \xi_7} - \frac{\eta}{1 - \eta_6} = 0 \quad (2.3)$$

On en déduit donc la fonction L_3 et de manière analogue les autres :

$$\left\{ \begin{array}{l} L_1(\xi, \eta) = (1 - \xi)(1 - \eta) \left(1 - \frac{\xi}{\xi_5} - \frac{\eta}{\eta_8} \right) \\ L_2(\xi, \eta) = \xi(1 - \eta) \left(\frac{\xi}{1 - \xi_5} - \frac{\eta}{\eta_6} - \frac{\xi_5}{1 - \xi_5} \right) \\ L_3(\xi, \eta) = -\xi\eta \left(\frac{\eta_6}{1 - \eta_6} + \frac{1}{1 - \xi_7} - \frac{\xi}{1 - \xi_7} - \frac{\eta}{1 - \eta_6} \right) \\ L_4(\xi, \eta) = (1 - \xi)\eta \left(\frac{\eta}{1 - \eta_8} - \frac{\xi}{\xi_7} - \frac{\eta_8}{1 - \eta_8} \right) \\ L_5(\xi, \eta) = \frac{1}{\xi_5(1 - \xi_5)} \xi(1 - \xi)(1 - \eta) \\ L_6(\xi, \eta) = \frac{1}{\eta_6(1 - \eta_6)} \xi\eta(1 - \eta) \\ L_7(\xi, \eta) = \frac{1}{\xi_7(1 - \xi_7)} \xi(1 - \xi)\eta \\ L_8(\xi, \eta) = \frac{1}{\eta_8(1 - \eta_8)} (1 - \xi)\eta(1 - \eta) \end{array} \right. \quad (2.4)$$

Si l'on prend les points $a_{i+4}|_{i=1\dots 4}$ au milieu de chacun des côtés ($\xi_i = \eta_i = \frac{1}{2}$), on retrouve les fonctions classiques d'interpolation des méthodes d'éléments finis (voir le livre [Dhatt 2007]). Pour construire l'image M d'un sommet \tilde{M} du maillage, on s'appuie sur les points les plus pertinents pour chaque point M , c'est-à-dire quand on adapte les points $a_{i+4}|_{i=1\dots 4}$ au point $\tilde{M} \begin{pmatrix} \xi \\ \eta \end{pmatrix}$ considéré, autrement dit quand $\xi_5 = \xi_7 = \xi$ et $\eta_6 = \eta_8 = \eta$. Il vient donc :

$$\left\{ \begin{array}{l} L_1(\xi, \eta) = -(1 - \xi)(1 - \eta) \\ L_2(\xi, \eta) = -\xi(1 - \eta) \\ L_3(\xi, \eta) = -\xi\eta \\ L_4(\xi, \eta) = -(1 - \xi)\eta \\ L_5(\xi, \eta) = (1 - \eta) \\ L_6(\xi, \eta) = \xi \\ L_7(\xi, \eta) = \eta \\ L_8(\xi, \eta) = (1 - \xi) \end{array} \right. \quad (2.5)$$

2^e étape : projection

Le maillage du domaine Ω est obtenu comme image de celui de $\tilde{\Omega}$. Les points isoparamétriques que l'on a estimés ne sont pas forcément situés sur la surface. Ceci vient du fait que l'on n'avait pas d'informations sur les variations de notre surface ailleurs que sur les bords. Il est donc nécessaire de projeter les points sur la surface morte et d'en extraire les coordonnées. Ces points sont portés par la surface à reconstruire mais pas forcément situés sur les lignes isoparamétriques. De même, il est possible d'étendre la méthode à un domaine topologiquement analogue à un triangle (cf. [George 1991]).

2.1.2.4 Création du réseau de courbes

L'objectif est de créer un réseau de courbes à partir du nuage de points précédemment construit. Les courbes utilisées sont généralement des splines dont on rappelle ci-dessous quelques

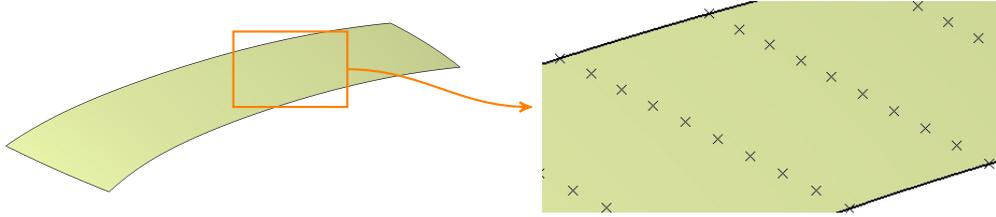


FIGURE 2.6 – Discrétisation de la surface de base

détails.

Soient :

- n : un entier tel que $n \geq 1$;
- V_0, V_1, \dots, V_n : $n + 1$ points de l'espace. V_i désigne soit un point de l'espace ($V_i = P_i$), soit une valeur associée à un point de l'espace (comme la dérivée au point P_i : $V_i = \dot{P}_i$) ;
- I : l'intervalle $[a, b]$, que l'on adimensionne ;
- t : un paramètre dont les valeurs $t_i|_{i=1..n}$ sont telles que $a = t_0 \leq \dots \leq t_n = b$ correspondent aux points V_0, V_1, \dots, V_n . On suppose que l'on a une répartition uniforme des t_i , ce qui permet d'alléger les calculs ;
- r : un ensemble de $n - 1$ nombres entiers vérifiant $0 < r_i \leq n$.

On souhaite construire une fonction polynomiale par morceaux de t et de classe C^{r_i-1} en t_i (*i.e.* au point V_i). La représentation $C(t)$ qui suit est facile à mettre en œuvre numériquement :

$$C(t) = [T][M][P] \tag{2.6}$$

avec $[T] = [t^n, t^{n-1}, \dots, t, 1]$ les polynômes de base de la représentation, $[M]$ une matrice de coefficients et $[P] = [V_0, V_1, \dots, V_n]^t$ l'ensemble des valeurs sur lesquelles s'appuie l'approximation.

La définition de la représentation d'une courbe se fait :

- soit *globalement* en utilisant les valeurs V_i données en une seule définition.
On utilise par exemple la forme de Lagrange (la courbe passe par les $n + 1$ points donnés), celle d'Hermite (on ajoute par rapport à Lagrange $n + 1$ dérivées aux points précédents ($\dot{P}_0, \dot{P}_1, \dots, \dot{P}_n$)) ou de Bézier (la courbe ne passe pas par les points de contrôle). Quand n est grand, les approximations précédentes conduisent soit à des oscillations, soit à des calculs prohibitifs ;
- soit *par morceaux* en utilisant pour chaque sous-courbe la forme permettant d'obtenir les propriétés requises au niveau de leur raccord. Chaque partie s'appuie sur un ensemble de valeurs pris parmi l'ensemble donné et son degré p est limité à une petite valeur ($p < 5$).

Les trois formes de la méthode globale (Lagrange, Hermite, Bézier) peuvent être transposées à la méthode par morceaux et enrichies pour améliorer les propriétés de raccords entre les sous-courbes. De nombreux exemples sont donnés dans la littérature ainsi que les valeurs de la matrice $[M]$ pour chaque cas (le lecteur pourra consulter le livre [George 1991] ou encore le chapitre 12 du livre [Frey 1999]).

La base cubique de béta-spline est une forme qui nous intéresse dans notre cas. Chaque sous-courbe est définie du point P_i au point P_{i+1} , mais utilise pour sa construction les points P_{i-1} et P_{i+2} . Deux paramètres sont introduits dans la matrice $[M]$: un paramètre β_1 de biais et un

paramètre β_2 de tension. On contrôle ainsi le comportement de la courbe en la déplaçant vers les points de contrôle.

$$[M] = \frac{1}{\Delta} \begin{pmatrix} -2\beta_1^3 & 2(\beta_2 + \beta_1^3 + \beta_1^2 + \beta_1) & -2(\beta_2 + \beta_1^2 + \beta_1 + 1) & 2 \\ 6\beta_1^3 & -3(\beta_2 + 2\beta_1^3 + 2\beta_1^2) & 3(\beta_2 + 2\beta_1^2) & 0 \\ -6\beta_1^3 & 6(\beta_1^3 - \beta_1) & 6\beta_1 & 0 \\ 2\beta_1^3 & \beta_2 + 4(\beta_1^2 + \beta_1) & 2 & 0 \end{pmatrix} \quad (2.7)$$

avec $\Delta = \beta_2 + 2\beta_1^3 + 4\beta_1^2 + 4\beta_1 + 2$.

Chaque bêta-spline s'appuie donc sur un ensemble de points par lesquels on impose son passage. À l'aide de ces points, on pourra ensuite agir sur la forme de la courbe.

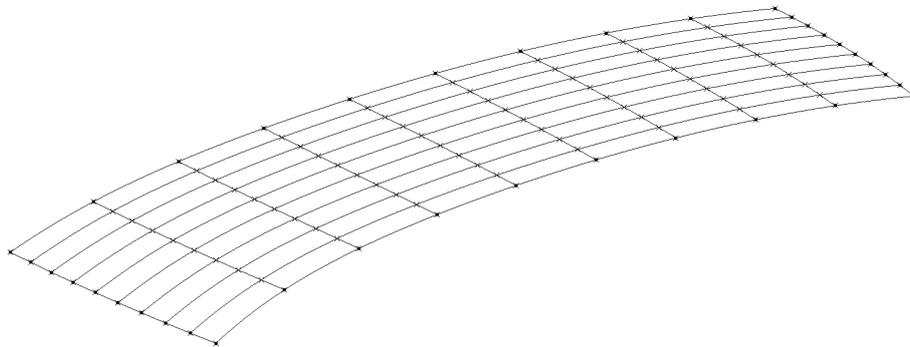


FIGURE 2.7 – Création du réseau de courbes

2.1.2.5 Création de la surface sur réseau

La surface à reconstruire s'appuie sur le réseau de courbes que l'on a créé auparavant. Pour cela, de nombreuses fonctions sont disponibles dans les logiciels CAO. Elles se basent sur un ensemble de profils et de courbes guides. Il est important de vérifier que l'on obtient bien les continuités voulues entre les différents patches (cf. annexe A.1 (p. 123)) au moyen des outils d'analyse disponibles dans les logiciels.

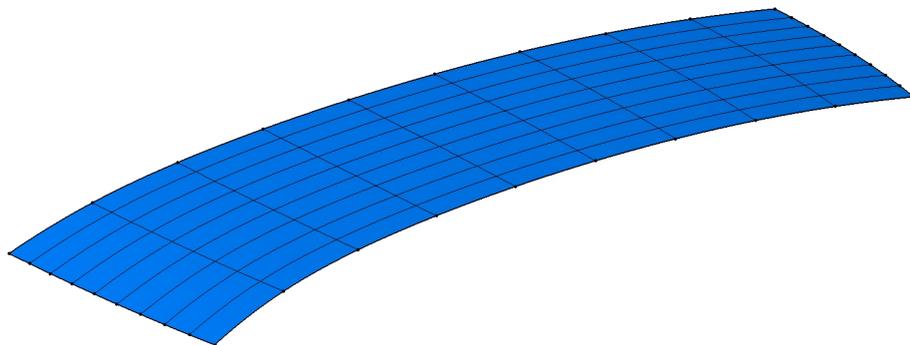


FIGURE 2.8 – Création de la surface sur réseau

A ce stade, notre surface morte est maintenant caractérisée par un réseau de courbes. Nous allons ensuite proposer une méthode pour gérer la déformation de cette grille.

2.1.3 Paramétrisation

On souhaite modifier notre surface en jouant sur des paramètres. Lorsque la valeur d'un paramètre est changée, le système répercute les modifications les unes après les autres [Gardan 2002]. Il faut donc introduire des paramètres (qui correspondent par exemple à des longueurs, des angles, des ratios) dans notre réseau de courbes. Ils sont issus de règles métier.

La surface est vue comme une nappe posée sur le réseau de courbes. Ces courbes sont des fils tendus passant par des points : il suffit de déplacer ces points pour déformer les courbes et donc la surface. On ne va toutefois pas piloter chaque nœud séparément car on risque de créer des surfaces bosselées. On préfère donc piloter tous les nœuds d'une même courbe selon une loi de déformation assez régulière comme un polynôme. Le nombre de contraintes va dicter le degré du polynôme.

Prenons l'exemple du pavillon du véhicule dans le plan (O,x,z) . On note $\Delta_z(x)$ la courbe déformée selon z au point de coordonnée x et δ la valeur de la déformation. On cherche à modifier la hauteur du point extrême haut ($\Delta_z(x_e) = \delta$) tout en y conservant une dérivée nulle ($\dot{\Delta}_z(x_e) = 0$) sans engendrer de déformation ni au bord d'attaque (c'est-à-dire du côté du pare-brise $\Delta_z(0) = 0$) ni au bord de fuite ($\Delta_z(L) = 0$). Ces quatre contraintes indépendantes font que l'on utilise une loi de déformation cubique $\Delta_z(x) = a_3x^3 + a_2x^2 + a_1x + a_0$. Les coefficients a_0 , a_1 , a_2 et a_3 sont alors facilement déterminés :

$$\begin{cases} a_0 = 0 \\ a_1 = \delta \frac{x_e L (2L - 3x_e)}{x_e^2 (L - x_e)^2} \\ a_2 = -\delta \frac{L^2 - 3x_e^2}{x_e^2 (L - x_e)^2} \\ a_3 = \delta \frac{L - 2x_e}{x_e^2 (L - x_e)^2} \end{cases}$$

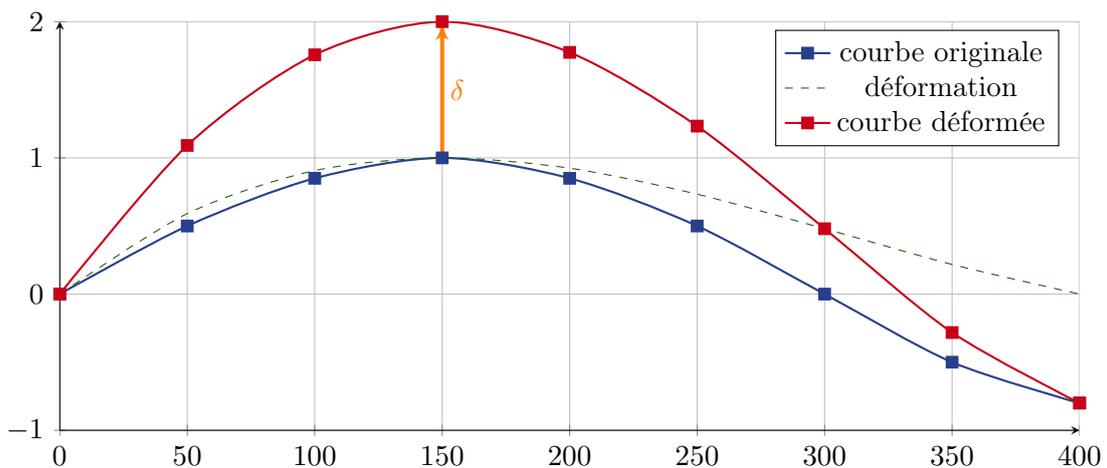


FIGURE 2.9 – Déformation de la courbe du pavillon en $y = 0$

La figure 2.10 présente de manière schématique notre méthode de reconstruction surfacique.

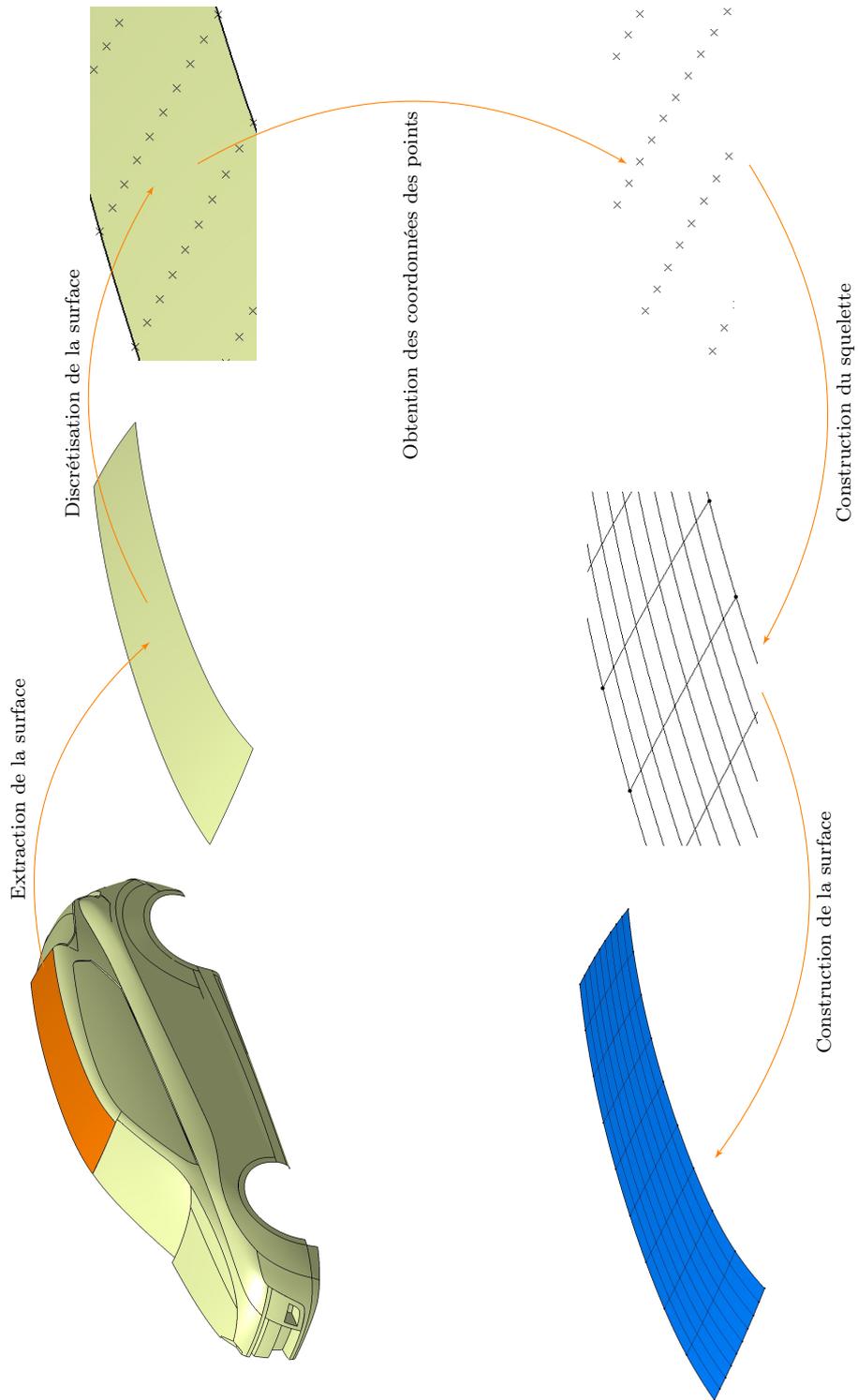


FIGURE 2.10 – Schéma de la reconstruction surfacique : exemple du toit de la Mégane iii (B95)

2.1.4 Application à un cas industriel

Afin de mettre en pratique ce qui précède, nous nous sommes appuyés sur le logiciel CATIA® V5 et son API basée sur le langage VB afin d’implémenter notre méthode. Cette tâche a été en grande partie automatisée et nous interférons avec le logiciel par le biais d’une IHM développée à cet égard. La partie de notre programme contient environ 10000 lignes de code informatique au total. Cette approche a été mise en œuvre sur différents designs de la gamme Renault. Ci-après sont présentées quelques résultats sur B98, plus connue sous le nom de Clio iv (voir la figure 2.11). Il faut environ 3 jours au total (c’est-à-dire pour le nettoyage et la reconstruction) pour générer un modèle entièrement paramétré d’un véhicule là où auparavant il n’existait pas de méthode dédiée et adaptée. La reconstruction proposée dans les figures 2.12 et 2.13 se fait en moins d’une journée lorsque la géométrie initiale a été pré-nettoyée. Le taux de reconstruction de ce modèle avoisine les 100 %.

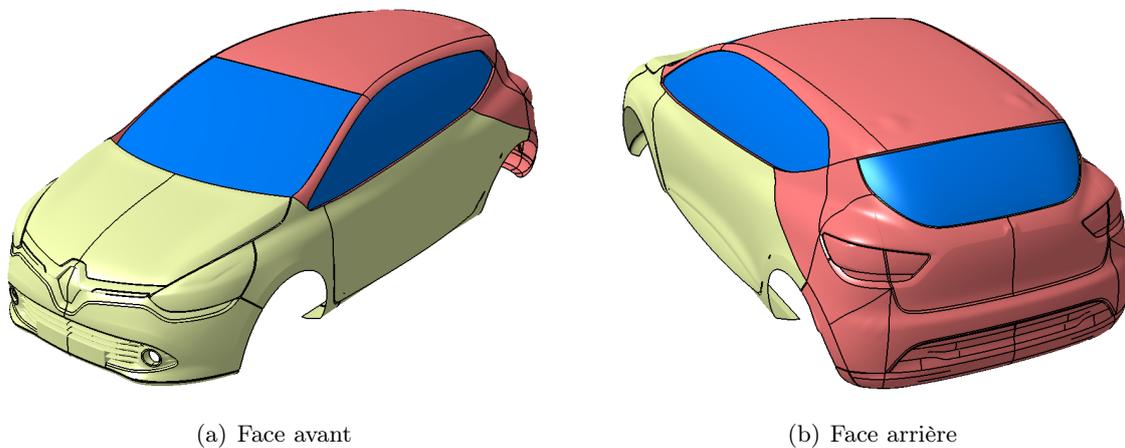


FIGURE 2.11 – Reconstruction surfacique de Clio iv : les surfaces en rouge correspondent aux parties reconstruites. Elles viennent se raccorder aux surfaces initiales de couleur sable.

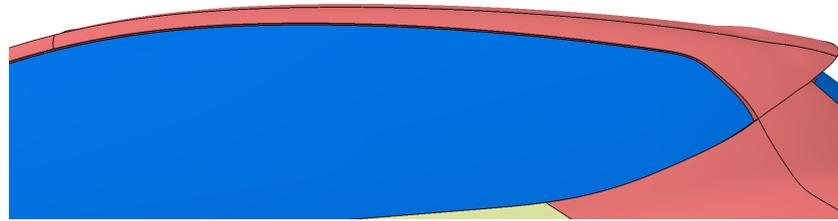
2.2 Analyse hors simulation d’un modèle paramétré

Chaque paramètre a sa plage de variation mais pour autant toutes les combinaisons ne sont pas réalisables. Il y a plusieurs raisons à cela :

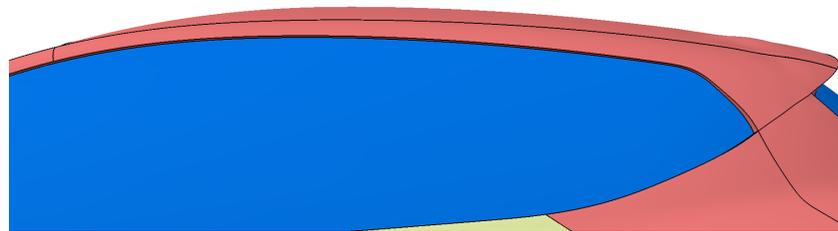
- soit parce qu’une combinaison peut s’avérer géométriquement infaisable : on parle alors de *zones interdites* (pas forcément connues à l’avance) ;
- soit parce que le modeleur présente certains bogues.

2.2.1 Détection des zones interdites

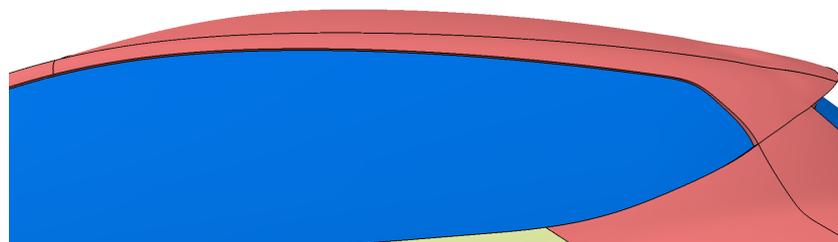
Lors de la génération des géométries, nous avons constaté que toutes les combinaisons de valeurs des paramètres ne sont pas faisables : l’espace de conception est rogné. Cette partie détaille une procédure simple qui permet d’analyser l’indépendance des paramètres du modèle CAO et de corriger certaines erreurs. Afin d’éclairer au mieux le lecteur, nous proposons un



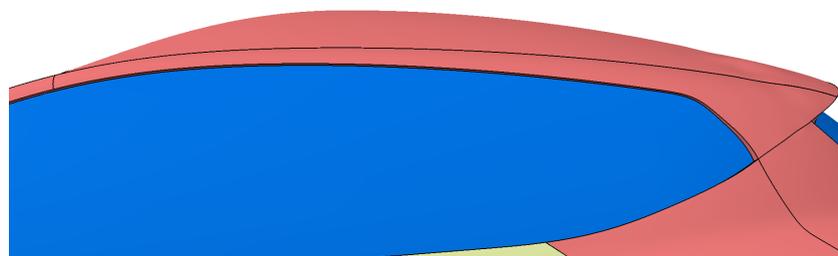
(a) Forme initiale



(b) $\delta = 30 \text{ mm}$



(c) $\delta = 60 \text{ mm}$



(d) $\delta = 100 \text{ mm}$

FIGURE 2.12 – Évolution de la flèche du pavillon

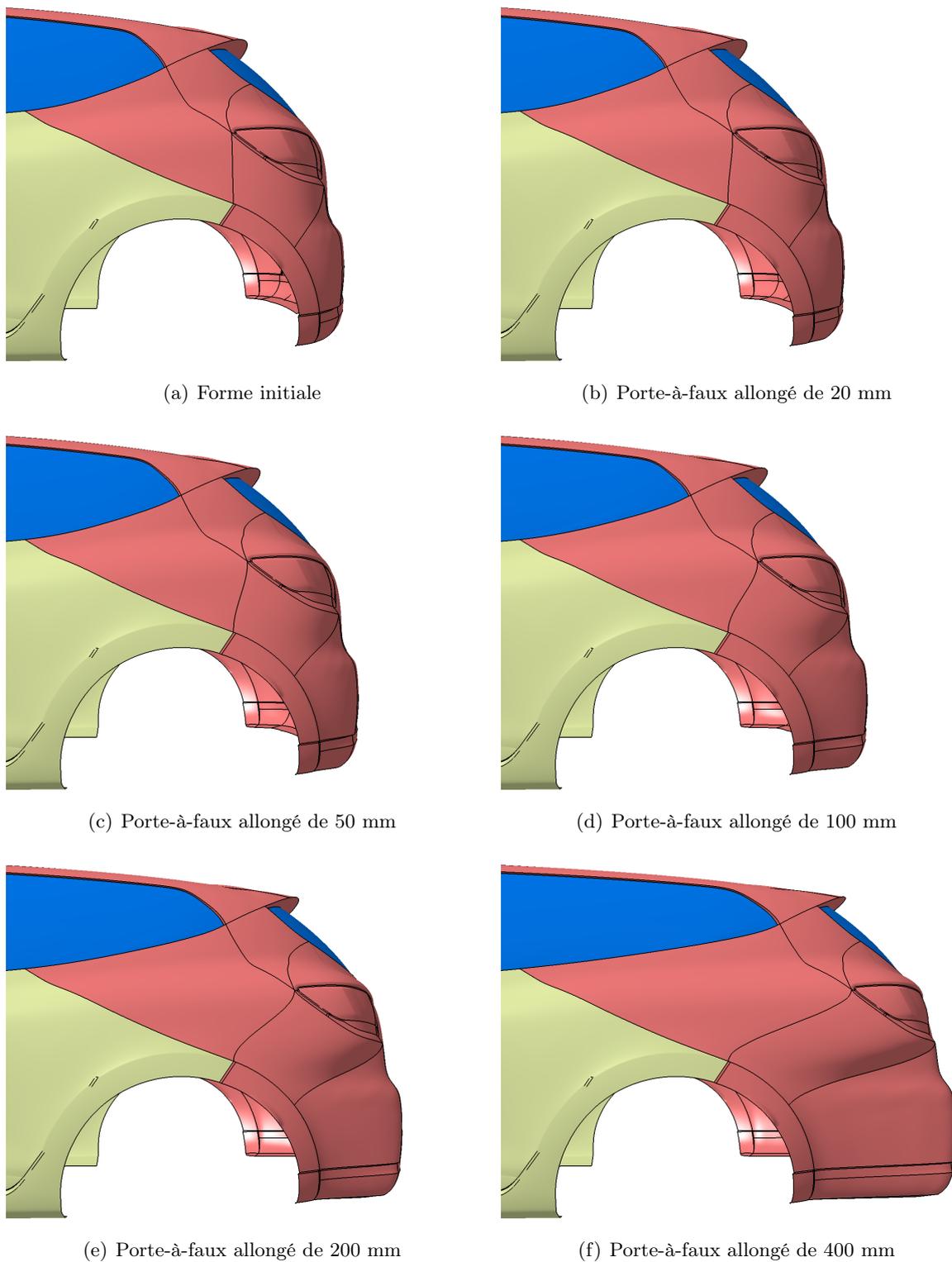


FIGURE 2.13 – Évolution du porte-à-faux arrière

exemple simple qui reflète les problèmes rencontrés dans les milieux industriels. On se donne une plaque percée dont la hauteur et les rayons des trous sont paramétrés (H , R_1 et R_2). Nous traçons un histogramme des essais réussis en fonction de chaque paramètre et aussi une répartition des succès et des échecs en fonction de chaque couple de paramètres possible.

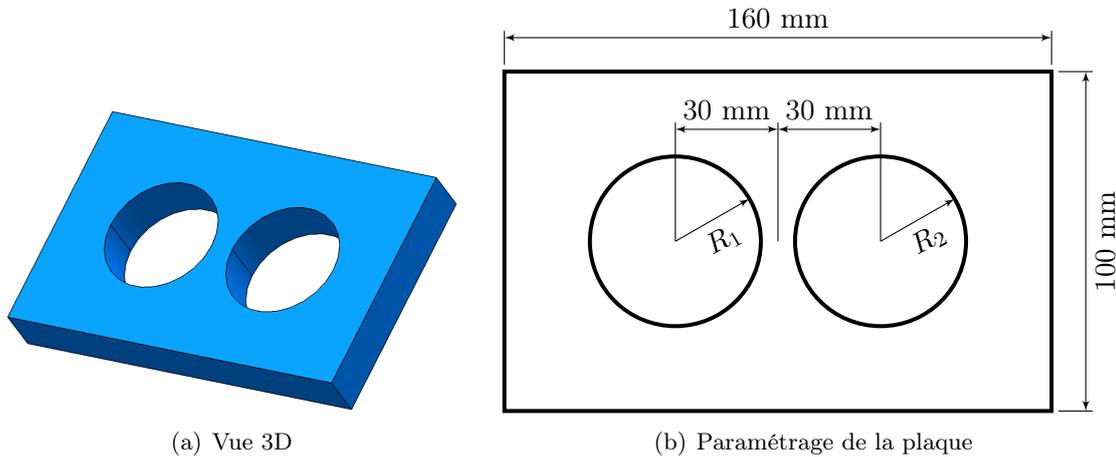


FIGURE 2.14 – Plaque percée

1^{er} cas : deux paramètres (H et R_1)

Dans ce cas, le paramètre R_1 varie entre 0 mm et 50 mm, R_2 vaut 25 mm et H varie entre 15 mm et 25 mm. 500 essais ont été tirés via un plan LHS (ces plans sont détaillés dans l'annexe B.1.2 (p. 126)). Les figures 2.15(a) et 2.15(b) ne nous apprennent pas grand chose. En effet, la géométrie n'est pas reconstituée lorsque les deux cercles se superposent, ce qui est totalement indépendant de la hauteur H . On observe simplement que le nombre de succès est logiquement quasiment constant sur le domaine de variation de H .

Comme on s'y attendait, la figure 2.15(c) nous confirme que la géométrie n'est pas reconstituée lorsque R_1 est compris entre 35 mm et 50 mm. Ceci est corroboré par l'histogramme 2.15(d) avec aucun essai réussi pour toutes les classes au delà de 35 mm. On trouve bien un taux d'échecs de 30 % : comme l'intervalle de variation de H_1 est découpé en 500 segments, les échecs correspondent au 150 de droite, *i.e.* au delà de 35 mm.

In fine, les différentes figures tracées permettent d'identifier facilement un problème de reconstruction et de mettre en lumière un paramètre dont la plage de variation serait trop grande. Le cas qui suit met en avant une interaction entre deux paramètres.

2^d cas : trois paramètres (H , R_1 et R_2)

Le paramètre R_2 varie cette fois entre 20 mm et 30 mm. Une géométrie n'est pas reconstituée lorsque $R_1 + R_2 > 60$ mm. Cette interaction entre les paramètres R_1 et R_2 se retrouve bien sur les figures 2.16(b) et 2.16(c) avec une séparation linéaire entre les succès et les échecs. Ceci est confirmé notamment par l'histogramme 2.16(a) où l'on observe une diminution progressive au delà de 30 mm du nombre de succès pour le paramètre R_1 . Dans une moindre mesure, l'histogramme 2.16(d) confirme également le propos précédent. Le taux d'échecs vaut environ 30.40 %.

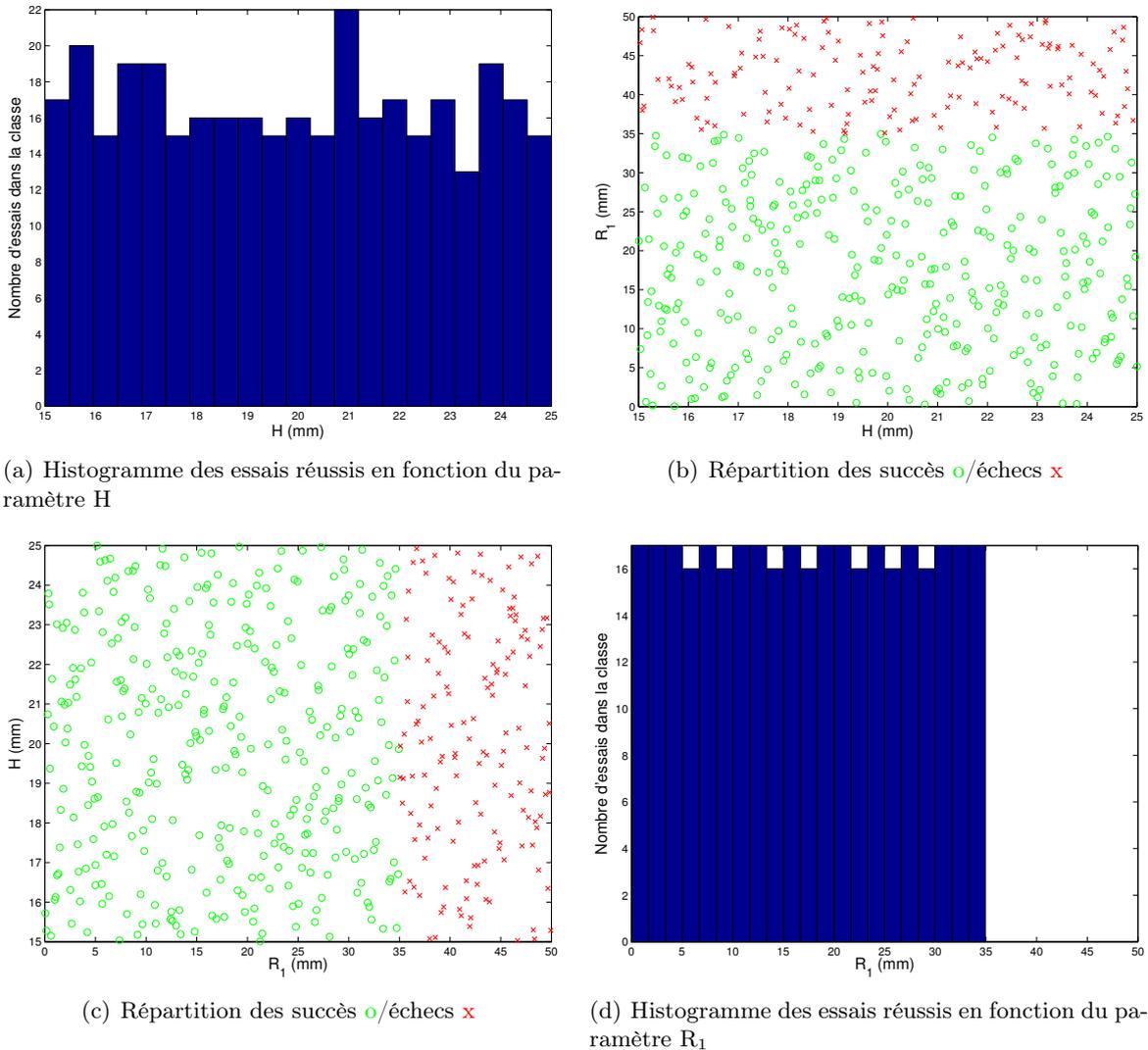


FIGURE 2.15 – Analyse des résultats avec 2 paramètres

Les autres figures (non représentées) qui font intervenir le paramètre H n’apportent évidemment aucune information.

Cette méthode permet donc de visualiser une interaction entre deux (en 2D) ou trois (en 3D) paramètres. Le concepteur est libre par la suite de diminuer les plages de variation des paramètres incriminés, ou bien d’intégrer une loi entre les paramètres dans le modèle CAO pour améliorer le taux de reconstruction. Un inconvénient de cette approche est qu’elle nécessite de tracer $\frac{p(p-1)}{2}$ figures lorsque l’on a p paramètres. Il serait intéressant de poursuivre le développement de cette méthode en automatisant le post-traitement de la lecture des graphiques, ce qui n’a pas été abordé dans cette thèse.

2.2. Analyse hors simulation d'un modèle paramétré

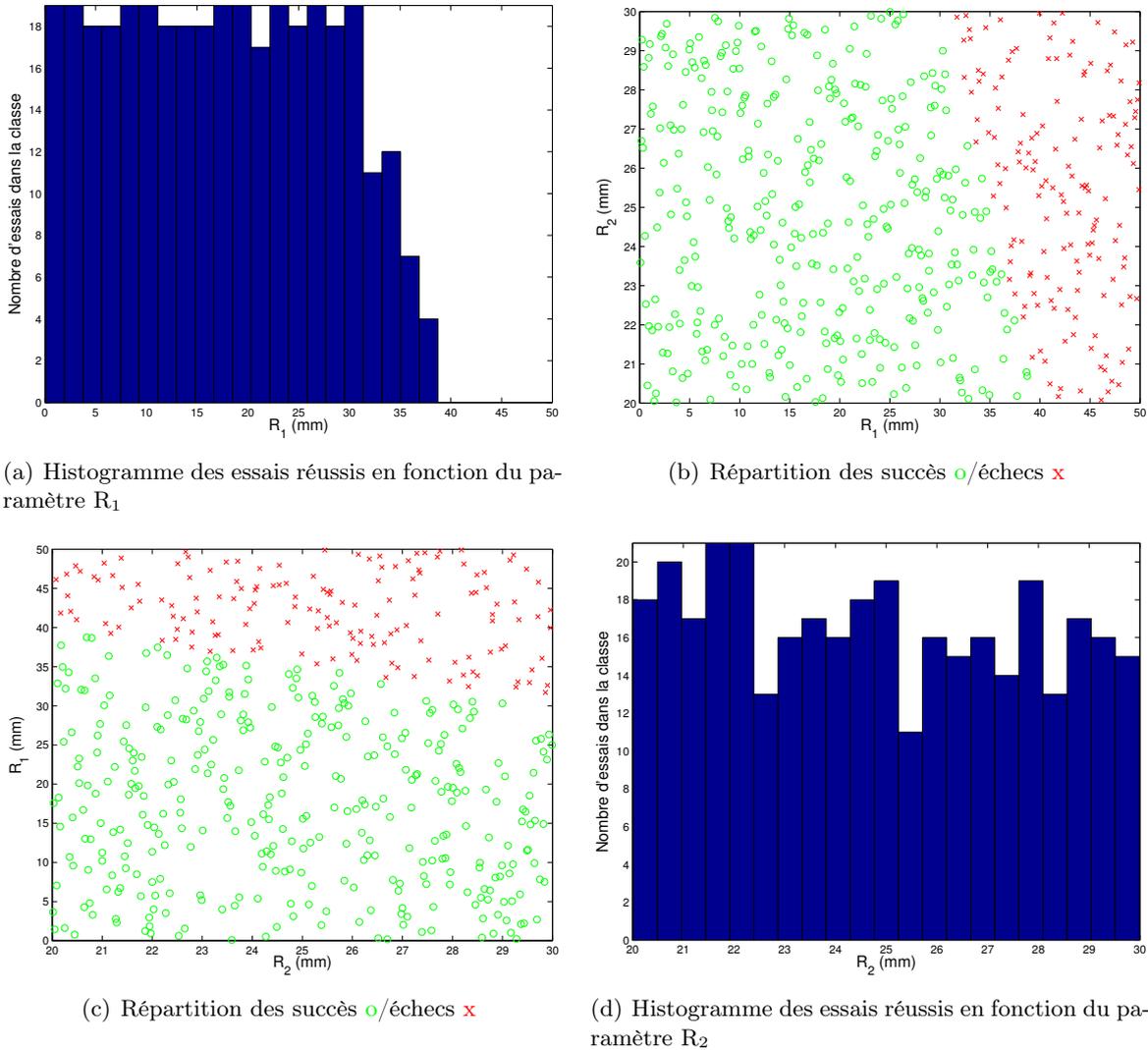


FIGURE 2.16 – Analyse des résultats avec 3 paramètres

2.2.2 Gestion des zones interdites par machine à vecteurs de support

Dans un contexte d'optimisation par plans d'expériences, deux approches sont possibles pour gérer les zones interdites :

- soit modéliser l'espace admissible ;
- soit générer une base conséquente d'essais candidats faisables et construire chaque nouveau plan d'expériences en se servant dans cette base (voir la partie 2.2.3 (p. 57)).

Comme on a pu le voir précédemment, un des problèmes inhérents aux modèles CAO est la connaissance de l'espace réel de conception, c'est-à-dire l'espace qui prend en compte les zones interdites du domaine d'étude. L'idée est de modéliser ce domaine par une frontière discriminant les « bons » et les « mauvais » essais. Pour classer les essais, nous utilisons des machines à vecteurs de support (de l'anglais *support vector machines*, abrégé en *SVM* par la suite). Géométriquement, on cherche l'hyperplan optimal qui sépare les essais selon leur classe d'appartenance.

Par *hyperplan optimal*, on entend celui qui se trouve « le plus loin possible de tous les essais ». Si l'on reprend l'exemple 2.16(b), l'hyperplan théorique est tracé sur la figure 2.17.

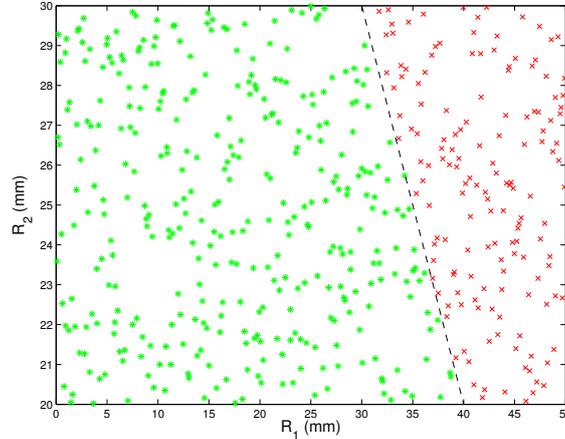


FIGURE 2.17 – Illustration en dimension 2 de l'hyperplan séparateur

Afin de modéliser la frontière du domaine, il est nécessaire de générer des essais sur l'ensemble de l'hypercube d'étude. C'est sur cette série d'essais que notre analyse discriminante va se baser pour prédire l'appartenance à l'une ou l'autre catégorie : *succès* ou *échecs*. On propose de tirer un plan hypercube latin qui permet d'assurer l'absence de redondance de l'information au travers d'une bonne répartition des projections sur les axes factoriels (cf. annexe B.1 (p. 125)). Puis, on génère les géométries avec notre logiciel CAO. Lorsque la géométrie est faisable, on la tague +1, autrement -1. Au final, nous avons un ensemble d'essais $x_i|_{i=1\dots n} \in \mathbb{R}^d$ dont on connaît l'appartenance à l'une ou l'autre catégorie ($y_i = \pm 1$).

La description de la théorie ci-après s'inspire de différentes publications comme [Boser 1992, Gunn 1998, Cristianini 2000] et des travaux internes à Renault [Albertelli 2005].

2.2.2.1 Formulation mathématique pour des données séparables

Problème primal

L'équation d'un hyperplan est $\langle w, x \rangle + b = 0$ avec $w \in \mathbb{R}^d$ et b un réel, dont le vecteur normal est $n = \frac{w}{\|w\|}$. La distance d'un point P à l'hyperplan est donnée par $d(x, H(w, b)) = \frac{|\langle w, x \rangle + b|}{\|w\|}$. En effet, soit P un point de l'espace et M un point quelconque appartenant à l'hyperplan. On a alors :

$$\begin{aligned}
 d(x, H(w, b)) &= |\langle x_P - x_M, n \rangle| \\
 &= \frac{1}{\|w\|} \left| \langle x_P, w \rangle - \underbrace{\langle x_M, w \rangle}_{=-b} \right| \\
 &= \frac{|\langle w, x_P \rangle + b|}{\|w\|}
 \end{aligned} \tag{2.8}$$

Un point (x_i, y_i) est bien classé si et seulement si $y_i (\langle w, x_i \rangle + b) > 0$. Comme l'équation d'un hyperplan est définie à un coefficient multiplicatif près, on s'impose de dimensionner w et b de sorte que les équations des deux hyperplans canoniques (s'appuyant sur les essais les plus proches de l'hyperplan H) soient $\langle w, x \rangle + b = -1$ et $\langle w, x \rangle + b = 1$. On appelle *marge* la distance du point le plus proche à l'hyperplan. Par conséquent, la marge vaut $\frac{1}{\|w\|}$. Maximiser la marge revient donc à minimiser $\|w\|$ sous contraintes. On obtient donc le problème quadratique avec des contraintes linéaires suivant :

$$\begin{cases} \min_{w,b} \frac{1}{2} \langle w, w \rangle \\ \text{s.t. } \forall i, y_i (\langle w, x_i \rangle + b) \geq 1 \end{cases} \quad (2.9)$$

Les *vecteurs de support* ou les *points de support* sont les x_i sur la frontière, *i.e.* ceux pour lesquels $y_i (\langle w, x_i \rangle + b) = 1$. La classe d'appartenance d'un nouvel essai z est donnée par $\text{sign}(\langle w, z \rangle + b)$.

Problème dual

Il est plus simple de résoudre le problème dual. Pour l'obtenir, on introduit les multiplicateurs (de Karush, Kuhn et Tucker) α_i pour chaque contrainte :

$$L_p(w, b, \alpha) = \frac{1}{2} \langle w, w \rangle - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - 1) \quad (2.10)$$

Nous concevons que le Lagrangien possède toutes les propriétés requises pour admettre un point selle. Le rapport technique [Albertelli 2005] renvoie vers les articles qui le montrent. Or, si l'on trouve un point selle $(\hat{w}, \hat{b}, \hat{\alpha})$ de L_p , alors (\hat{w}, \hat{b}) est solution de notre problème. Soit la fonction duale $L_d(\alpha) = \min_{w,b} L_p(w, b, \alpha)$ que nous cherchons à maximiser :

$$(\hat{w}, \hat{b}, \hat{\alpha}) = \arg \max_{\alpha} L_d(\alpha) = \arg \max_{\alpha} \min_{w,b} L_p(w, b, \alpha) \quad (2.11)$$

Pour la fonction duale, les conditions d'optimalité nous donnent :

$$\begin{cases} \nabla_w L_p(w, b, \alpha) = w(\alpha) - \sum_i \alpha_i y_i x_i = 0 \\ \nabla_b L_p(w, b, \alpha) = -\sum_i \alpha_i y_i = 0 \end{cases} \Rightarrow \begin{cases} w(\alpha) = \sum_i \alpha_i y_i x_i \\ \sum_i \alpha_i y_i = 0 \end{cases} \quad (2.12)$$

Par substitution des résultats de l'équation (2.12) dans la fonction duale L_d , il vient :

$$\begin{aligned} L_d(\alpha) &= \frac{1}{2} \langle w, w \rangle - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - 1) \\ &= \frac{1}{2} \langle w, w \rangle - \underbrace{\sum_i \alpha_i y_i \langle w, x_i \rangle}_{=\langle w, w \rangle} - \underbrace{b \sum_i \alpha_i y_i}_{=0} + \sum_i \alpha_i \\ &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \end{aligned} \quad (2.13)$$

On obtient donc le problème quadratique dual suivant :

$$\begin{cases} \hat{\alpha} = \arg \max_{\alpha \geq 0} \left[\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \right] \\ s.t. \sum_i \alpha_i y_i = 0 \end{cases} \quad (2.14)$$

Seuls les α_i correspondant aux points les plus proches sont non nuls, on parle alors de *points de support*. Les α_i non nuls dans la solution du problème dual définissent l'hyperplan en donnant \hat{w} comme la somme des $\hat{\alpha}_i y_i x_i$ (cf. équation (2.12)). \hat{b} est obtenu par l'équation $y_i (\langle w, x_i \rangle + b) = 1$ en prenant un vecteur de support, *i.e.* en choisissant un i tel que $\alpha_i \neq 0$.

2.2.2.2 Formulation mathématique pour des données non séparables

Il n'est pas toujours possible de trouver un hyperplan séparateur, notamment parce que les données peuvent comporter du bruit. On utilise donc une notion de marge qui signifie qu'un hyperplan peut séparer la plupart des points, mais pas tous. On introduit des variables d'écart s_i , permettant de relâcher la contrainte, et une constante de pénalisation C . En l'absence de pénalisation, il y a un risque de ne pas trouver de solution admissible. Le problème primal devient donc :

$$\min_{w,b,s} \left(\frac{1}{2} \langle w, w \rangle + C \sum_i s_i \right) \quad (2.15)$$

tel que :

$$\begin{cases} \forall i, y_i (\langle w, x_i \rangle + b) \geq 1 - s_i \\ \forall i, s_i \geq 0 \end{cases} \quad (2.16)$$

Problème dual

En introduisant des multiplicateurs de Lagrange μ_i , la fonction à minimiser est :

$$L_{p'}(w,b,s,\alpha,\mu) = \frac{1}{2} \langle w, w \rangle + C \sum_i s_i - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - (1 - s_i)) - \sum_i \mu_i s_i \quad (2.17)$$

avec $\alpha_i \geq 0$ les multiplicateurs de Karush, Kuhn et Tucker associés aux contraintes $y_i (\langle w, x_i \rangle + b) - (1 - s_i) \geq 0$ et μ_i ceux associés aux contraintes $s_i \geq 0$. Le Lagrangien possède toutes les propriétés de convexité/concavité requises pour avoir un point selle (consulter [Albertelli 2005] pour plus d'informations). Soit la fonction duale $L_{d'}(\alpha,\mu) = \min_{w,b,s} L_{p'}(w,b,s,\alpha,\mu)$. Les conditions nécessaires pour avoir un point selle de $L_{p'}$ deviennent :

$$\begin{cases} \nabla_w L_{p'}(w,b,s,\alpha,\mu) = w(\alpha,\mu) - \sum_i \alpha_i y_i x_i = 0 \\ \nabla_b L_{p'}(w,b,s,\alpha,\mu) = - \sum_i \alpha_i y_i = 0 \\ \nabla_{s_i} L_{p'}(w,b,s,\alpha,\mu) = C - \alpha_i - \mu_i = 0 \end{cases} \Rightarrow \begin{cases} w(\alpha,\mu) = \sum_i \alpha_i y_i x_i \\ \sum_i \alpha_i y_i = 0 \\ \alpha_i = C - \mu_i \end{cases} \quad (2.18)$$

Par substitution des résultats de l'équation (2.18) dans la fonction duale L_d , il vient :

$$\begin{aligned}
 L_d &= \frac{1}{2} \langle w, w \rangle + C \sum_i s_i - \sum_i \alpha_i (y_i (\langle w, x_i \rangle + b) - (1 - s_i)) - \sum_i \mu_i s_i \\
 &= \frac{1}{2} \langle w, w \rangle - \underbrace{\sum_i \alpha_i y_i \langle w, x_i \rangle}_{=\langle w, w \rangle} - b \underbrace{\sum_i \alpha_i y_i}_{=0} + C \underbrace{\sum_i s_i + \sum_i \alpha_i (1 - s_i) - \sum_i \mu_i s_i}_{=\sum_i \alpha_i} \\
 &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle
 \end{aligned} \tag{2.19}$$

On obtient donc le problème quadratique dual qui suit :

$$\left\{ \begin{array}{l}
 (\hat{\alpha}, \hat{s}) = \arg \max_{\alpha \geq 0, s \geq 0} \left[\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \right] \\
 s.t. \left\{ \begin{array}{l}
 \sum_i \alpha_i y_i = 0 \\
 \forall i, \alpha_i = C - \mu_i
 \end{array} \right.
 \end{array} \right. \tag{2.20}$$

qui se réécrit après simplification :

$$\left\{ \begin{array}{l}
 \hat{\alpha} = \arg \max_{0 \leq \alpha \leq C} \left[\sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \right] \\
 s.t. \sum_i \alpha_i y_i = 0
 \end{array} \right. \tag{2.21}$$

La constante de pénalisation^b conserve les valeurs admissibles des multiplicateurs α_i dans une « boîte » ($0 \leq \alpha_i \leq C$). Augmenter C donne plus de poids aux variables d'écart s_i , ce qui signifie que l'optimisation tente d'établir une séparation plus stricte entre les classes. Au contraire, une pénalisation trop faible induit que l'on explique trop grossièrement la relation entre les données et leur classe (phénomène de *sur-généralisation*). La figure 2.20 montre l'influence de la pénalisation sur le jeu de données précédent.

2.2.2.3 Transformations non linéaires

Certains problèmes de classification binaire n'admettent pas de simples hyperplans séparateurs. Il existe donc une variante mathématique qui se base sur la théorie des noyaux et qui conserve la quasi-totalité de la simplicité d'une approche SVM. On appelle *noyau* toute fonction de la forme $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$. Toute fonction découlant d'un produit scalaire est considérée comme un noyau. Cette notion de noyau peut être élargie aux fonctions vérifiant la condition de Mercer. On renvoie à la littérature, notamment [Collobert 2000], pour plus de détails. On peut citer à titre d'exemple :

- les noyaux polynomiaux (voir la figure 2.19) :

$$k(x, y) = (1 + \langle x, y \rangle)^k \tag{2.22}$$

b. La constante de pénalisation C est parfois appelée *contrainte de boîte*.

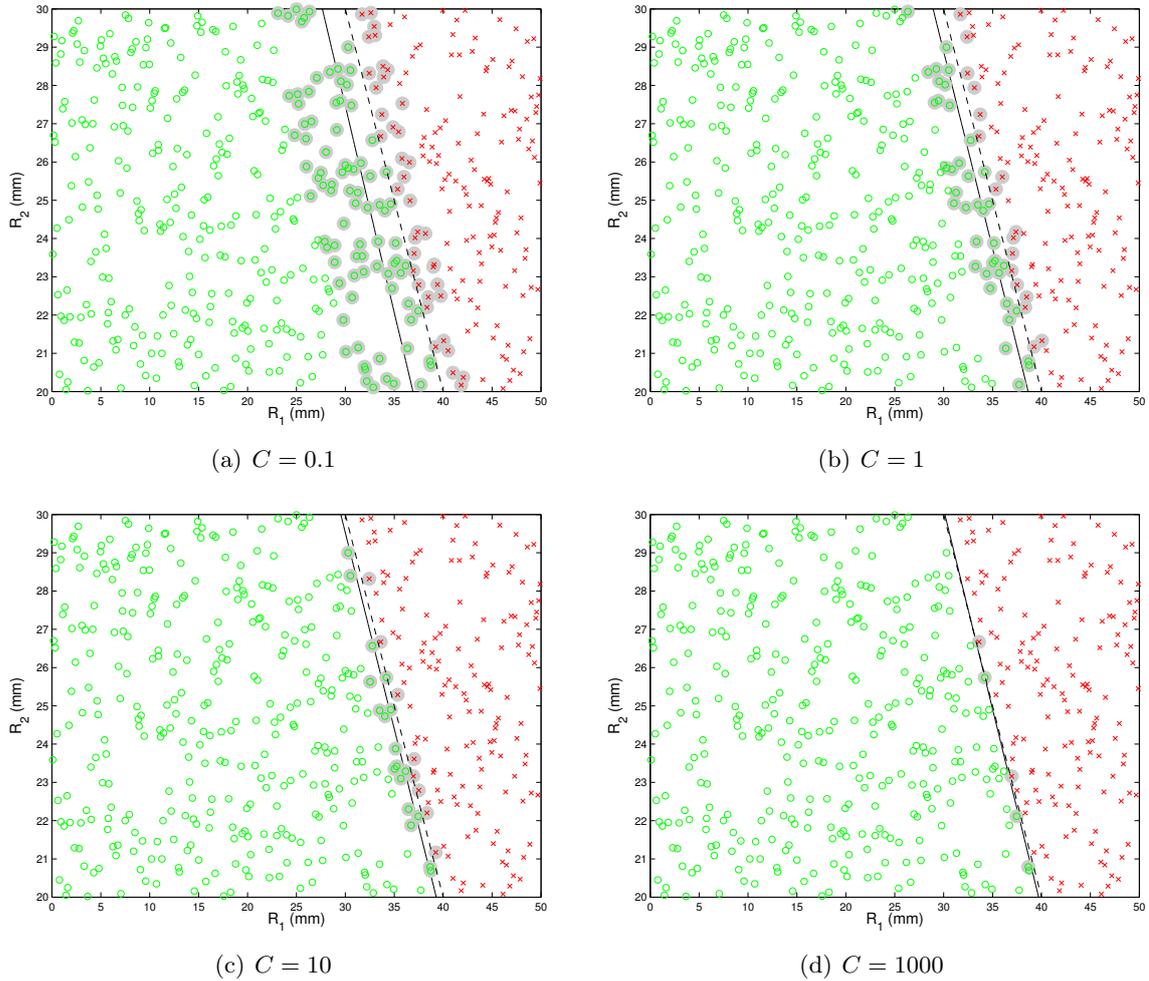


FIGURE 2.18 – Influence de la constante de pénalisation (500 essais, succès \circ , échecs \times , points de support \bullet , hyperplan séparateur calculé —, hyperplan séparateur théorique - -)

o les noyaux gaussiens (fonctions de base radiale) :

$$k(x,y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{2.23}$$

où σ est un paramètre ajustable. Pour éviter un phénomène de sur-apprentissage, σ ne doit pas être pris trop petit (voir la figure 2.20(a)). Pour éviter une sur-généralisation, il ne doit pas être pris trop grand (voir la figure 2.20(d)). En pratique, il est courant d'utiliser une validation croisée pour estimer la fiabilité du modèle et en déduire les meilleurs C et σ .

Une question reste en suspens : combien d'essais sont nécessaires en dimension d pour que la fiabilité de notre modèle soit bonne? Il n'y a pas de réponse toute faite. On ajustera de façon empirique le nombre d'essais nécessaires pour nos problèmes.

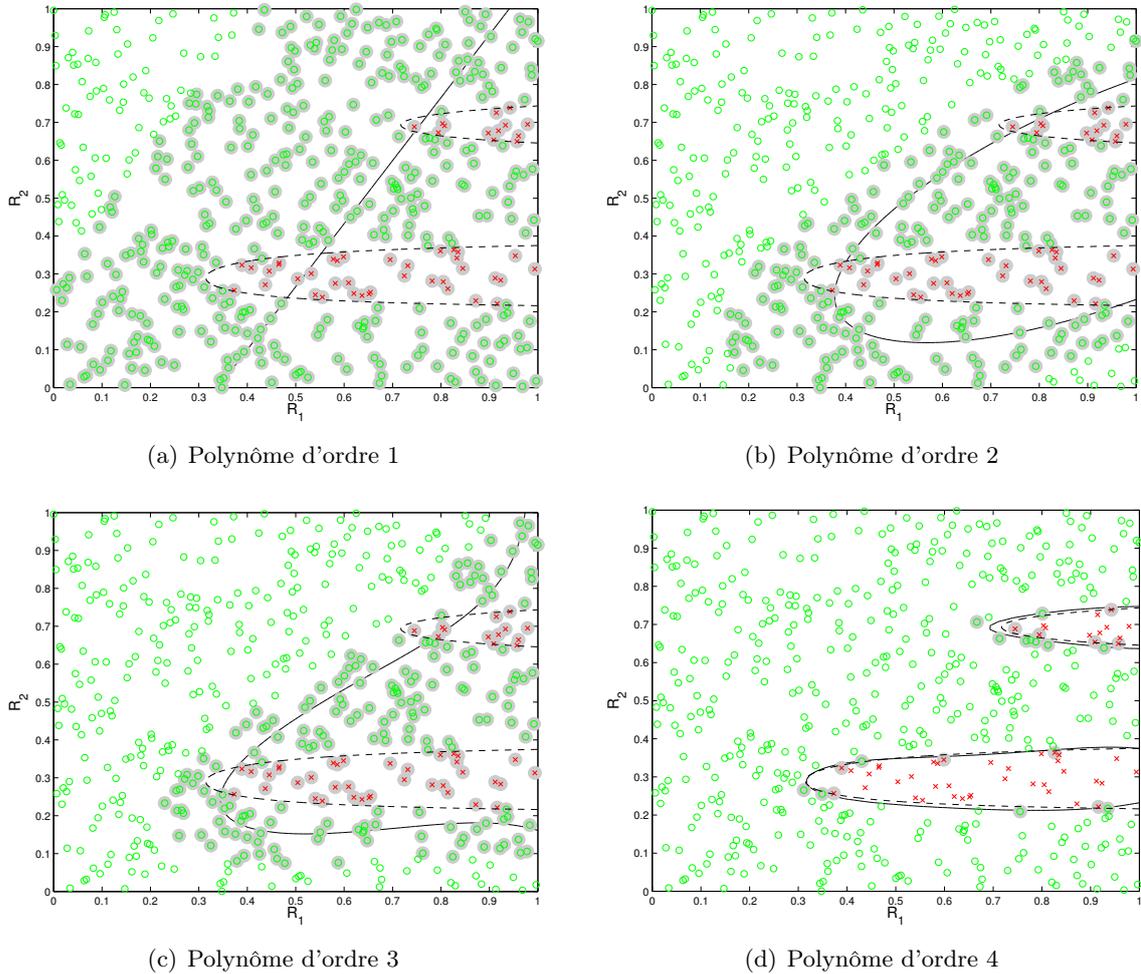


FIGURE 2.19 – Influence de l'ordre du polynôme (500 essais, succès \circ , échecs \times , points de support \bullet , hypersurface séparatrice calculée —, hypersurface séparatrice théorique - - -)

Enfin, il arrive parfois que l'on n'a pas lancé assez d'essais pour discriminer correctement la frontière. Il est donc opportun de compléter le plan d'expériences initial par de nouveaux essais. Nous proposons une méthode dans l'annexe B.2.4 (p. 130).

2.2.3 Gestion des zones interdites par génération d'un grand nombre d'essais

Cette méthode consiste à générer un grand nombre d'essais candidats faisables sur lesquels on s'appuiera pour construire un plan d'expériences. Toutefois, cette méthode brute peut s'avérer extrêmement coûteuse si le modèle CAO est complexe. De surcroît, une question demeure : « combien faut-il d'essais pour trouver un optimum acceptable ? ». Les performances de l'algorithme d'optimisation dépendent du nombre de candidats en entrée.

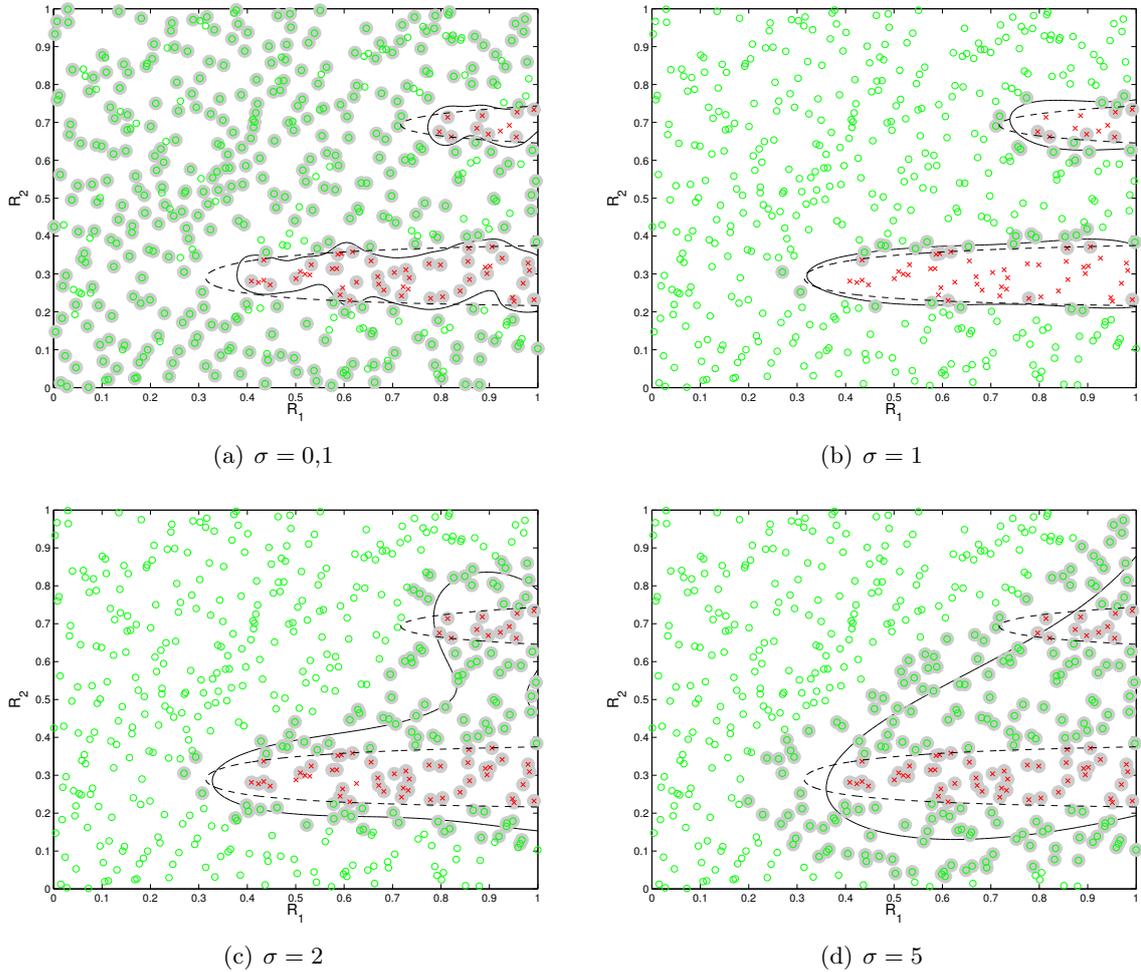


FIGURE 2.20 – Influence de σ (500 essais, succès \circ , échecs \times , points de support \bullet , hypersurface séparatrice calculée —, hypersurface séparatrice théorique - - -)

2.3 Réduction du nombre de paramètres *a priori*

On rappelle que les paramètres du modèle CAO sont les paramètres à optimiser. Ils nous donnent la dimension de l'espace dans lequel il faut chercher un optimum. Plus l'espace est grand, plus il faut d'essais. Or, dans le cadre de l'aérodynamique d'un véhicule, on peut compter plusieurs dizaines voire centaines de paramètres et chaque calcul peut durer plus de trois jours sur 64 cœurs! Il y a donc un gros intérêt à réduire le nombre de degrés de liberté et à limiter le coût d'un calcul (cf. partie 2.3.2 (p. 67)). Cette partie étudie deux techniques de réduction de paramètres : l'analyse en composantes principales et les réseaux de neurones artificiels.

2.3.1 1^{re} voie : voxelisation et analyse en composantes principales

Bien que l'on explore un espace de dimension m , il se peut que l'ensemble des réalisables soit une variété de dimension inférieure (par exemple, une surface dans un cube). Ceci peut

s'expliquer par un grand nombre de règles ou de zones interdites mais c'est difficile à détecter. C'est l'objet de cette partie.

2.3.1.1 Voxellisation

Pour déterminer la vraie dimension de notre problème, on va utiliser les analyses en composantes principales pour extraire les modes les plus importants (possiblement tous!). Mais auparavant, on va décrire les géométries par un autre format. On introduit une autre représentation géométrique basée sur une *pixellisation* en 2D ou *voxellisation* en 3D (cf. [Raghavan 2013]). Nous utilisons une grille de référence dont la résolution est prédéterminée et nous y plongeons l'ensemble des formes générées (cf. figure 2.21). En d'autres termes, nous représentons chaque forme Ω_i par une fonction indicatrice $\chi(x)$ en notant x une cellule (pixel ou voxel).

$$\begin{aligned} \chi(x) &= 1 \text{ si } x \in \bar{\Omega}_i \\ &= 0 \text{ si } x \notin \bar{\Omega}_i \end{aligned} \tag{2.24}$$

Ainsi, chaque forme^c S_i est représentée de manière unique et sans paramètre de forme par un vecteur binaire qui vaut 1 si la cellule est intersectée ou à l'intérieur du domaine et 0 autrement : $S_i \in \mathbb{R}^{n_c}$ où n_c est la résolution de la grille. Au final, on stocke l'ensemble des vecteurs binaires dans une matrice notée X , à n lignes et p colonnes (nombre de voxels).

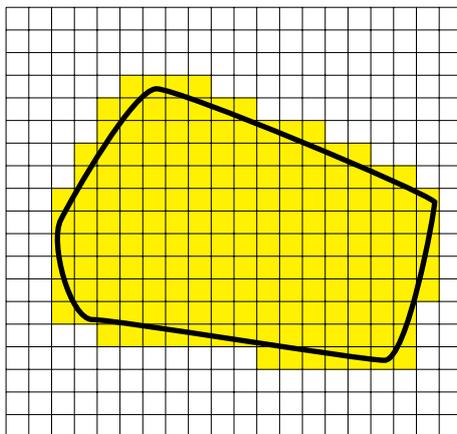


FIGURE 2.21 – Pixellisation d'une forme quelconque

2.3.1.2 Analyse en composantes principales

Cette méthode vise à transformer des variables corrélées en nouvelles variables décorrélées les unes des autres dites *composantes principales* ou *axes principaux*. *In fine*, cela permet de réduire le nombre de variables et de rendre l'information moins redondante.

Le point d'entrée de la méthode est une matrice X à n lignes (nombre de réalisations) et p colonnes (nombre de paramètres), de terme général x_{ij} . Si les n points représentant le tableau X dans cet espace sont contenus dans un sous-espace vectoriel de dimension q inférieur à p , il

c. Les formes sont couramment appelées « snapshots » ([Chatterjee 2000]).

est alors possible de reconstituer les positions des n points à partir des coordonnées sur les q nouveaux axes et des composantes de ces nouveaux axes.

On cherche un espace de dimension $q \ll p$ conservant les mêmes distances entre les points. Soit (v_i) la base de cet espace. La projection de X sur cet espace s'écrit :

$$\tilde{X} = \sum_{i=1}^q u_i v_i' \tag{2.25}$$

et $\|\tilde{X} - X\|$ l'erreur commise. Ainsi, on est capable de reconstituer les $n \cdot p$ valeurs de X avec $q \cdot (n + p)$ valeurs numériques seulement [Lebart 1982].

Pour mesurer les erreurs commises, on peut regarder la qualité globale de la reconstitution qui s'appuie sur les valeurs propres λ_α :

$$\tau_q = \frac{\sum_{\alpha=1}^q \lambda_\alpha}{\sum_{\alpha=1}^p \lambda_\alpha} \tag{2.26}$$

En résumé, on souhaite retenir les q premiers vecteurs propres en espérant que q soit largement inférieur au nombre de paramètres CAO initiaux.

1^{er} exemple : relations linéaires entre les données

On contraint un jeu d'essais contenant 2000 points dans un espace à 5 dimensions. Les deux premières dimensions sont indépendantes et les trois dernières sont des combinaisons linéaires des deux premières dimensions. Le problème a donc en réalité deux dimensions. C'est ce que nous voulons retrouver grâce à notre méthode.

L'analyse en composantes principales permet de retrouver la vraie dimension du problème, ce que l'on voit dans le tableau 2.1. En effet, les deux premiers vecteurs propres nous permettent de reconstruire fidèlement les données initiales.

N°	Valeur propre λ_α	Qualité globale de reconstruction τ_q
1	2.4622	0.90321
2	0.26384	1
3	0	1
4	0	1
5	0	1

Tableau 2.1 – Valeurs propres et qualité globale de reconstruction du plan avec des relations linéaires entre les données

2^e exemple : relations non linéaires entre les données

On remplace dans le jeu de données précédent les relations linéaires *par des relations quadratiques et cubiques*.

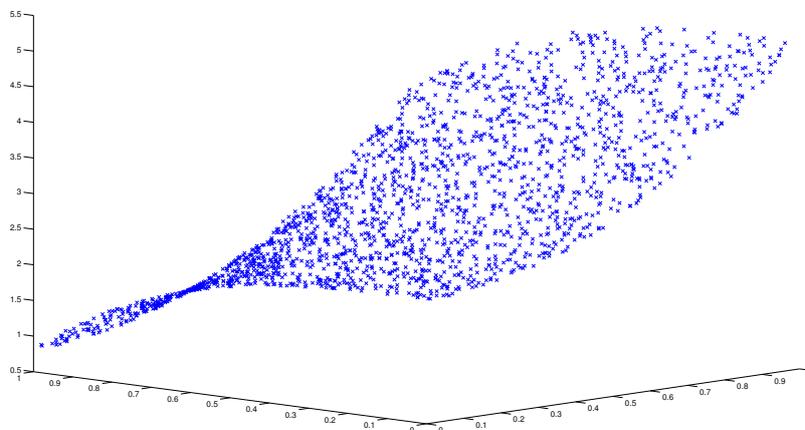


FIGURE 2.22 – Plan généré (avec 2000 candidats) en dimension 3 qui utilise des relations non linéaires entre les données

N°	Valeur propre λ_α	Qualité globale de reconstruction τ_q
1	3.5029	0.68888
2	1.5642	0.99649
3	0.012941	0.99903
4	0.0049243	1
5	0	1

Tableau 2.2 – Valeurs propres et qualité globale de reconstruction du plan avec des relations non linéaires entre les données

Le processus fait ressortir deux valeurs propres fortes (cf. tableau 2.2). Les deux vecteurs propres associés à ces deux valeurs propres constituent la base orthonormée à deux dimensions s’ajustant au mieux (au sens des moindres carrés) aux données. Toutefois, les valeurs propres ne sont qu’un estimateur global de la dimension intrinsèque des données, tout comme le maximum de vraisemblance n’est qu’un estimateur local, comme on peut le voir sur la figure 2.23.

Le tracé sur la figure 2.24 des coordonnées des points sur les trois premiers axes (*i.e.* les composantes de Xu_α avec $\alpha = 1..3$) confirme le fait que le jeu de données de départ est de dimension 2.

2.3.1.3 Exemple d’application

Description du modèle géométrique

On se donne un tube (cf. figure 2.25(a)) utilisant plus de paramètres que nécessaire pour générer la même forme. Son épaisseur est fixe. Le cercle central a la même origine que le repère

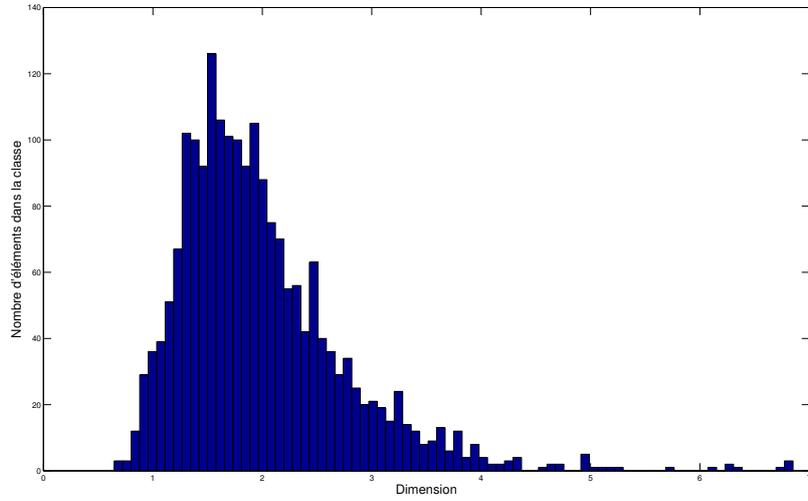


FIGURE 2.23 – Estimation de la dimension du problème par le maximum de vraisemblance

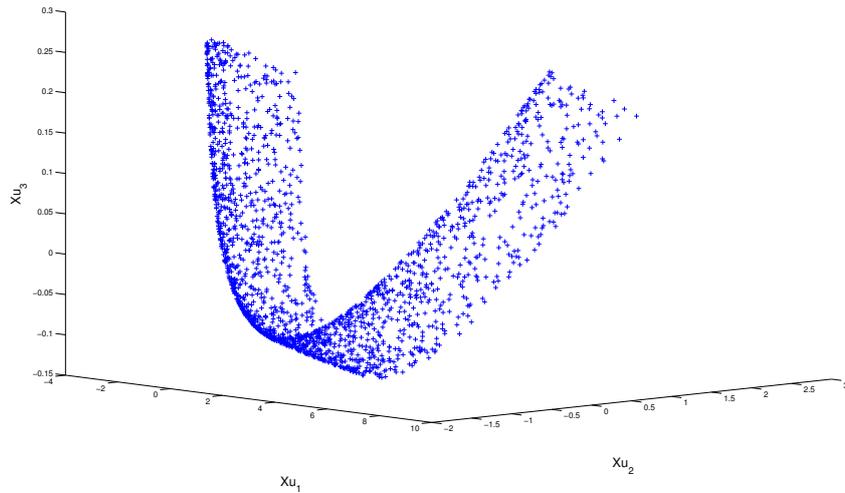


FIGURE 2.24 – Représentation des coordonnées des points sur les trois premiers axes

et a un rayon fixe. Le cercle externe est construit de façon à ce qu'il passe par 3 points P_1 , P_2 et P_3 et à ce que son centre P_0 soit situé sur l'axe des ordonnées (cf. figure (2.25(b))).

$$\begin{cases} P_0 = (0, y_0) \\ P_1 = (x_1, y_1) \\ P_2 = (x_2, y_2) \\ P_3 = (x_3, y_3) \end{cases} \quad (2.27)$$

Le tube est donc défini dans un environnement CAO par 5 paramètres : x_1 , y_1 , x_2 , y_2 et y_3 . Le paramètre x_3 dépend des autres paramètres de façon à ce que le centre du cercle externe se

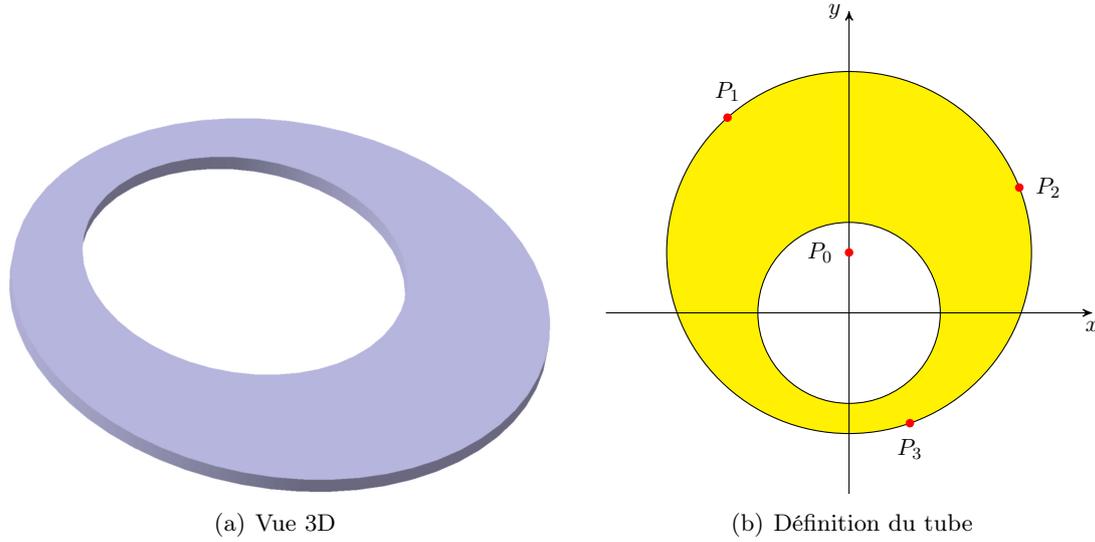


FIGURE 2.25 – Tube

situe toujours sur l'axe des ordonnées. On a le système d'équations suivant :

$$x_1^2 + (y_1 - y_0)^2 = R^2 \quad (2.28a)$$

$$x_2^2 + (y_2 - y_0)^2 = R^2 \quad (2.28b)$$

$$x_3^2 + (y_3 - y_0)^2 = R^2 \quad (2.28c)$$

avec y_0 , R et x_3 les inconnues. Il vient :

$$(2.28a) - (2.28b) \Rightarrow x_1^2 - x_2^2 + y_1^2 - y_2^2 - 2y_0(y_1 - y_2) = 0 \quad (2.29a)$$

$$(2.28b) - (2.28c) \Rightarrow x_2^2 - x_3^2 + y_2^2 - y_3^2 - 2y_0(y_2 - y_3) = 0 \quad (2.29b)$$

Ainsi :

$$(2.29a) \Rightarrow y_0 = \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2}{2(y_1 - y_2)} \quad (2.30a)$$

$$(2.29b) \Rightarrow x_3^2 = x_2^2 + y_2^2 - y_3^2 - 2y_0(y_2 - y_3) \quad (2.30b)$$

On a construit 5 degrés de liberté mais le problème est en réalité de dimension 2 car nos trois points P_1 , P_2 et P_3 se trouvent sur un cercle. La vraie dimension du problème pourrait se résumer aux paramètres suivant :

- l'ordonnée y_0 du centre du cercle externe ;
- le rayon R du cercle externe.

On note que le taux de reconstruction de cette géométrie est d'environ 11,52 % à cause de la zone interdite qui correspond à l'intersection des deux cercles quand $y_0 - r < R$ (avec r le rayon du cercle intérieur).

Voxellisation

On génère 250 géométries que l'on voxellise ensuite. On obtient donc un vecteur binaire pour chaque géométrie que l'on stocke dans une matrice notée X à 250 lignes (nombre de réalisations conjointes) et 554040 colonnes (nombre de voxels).

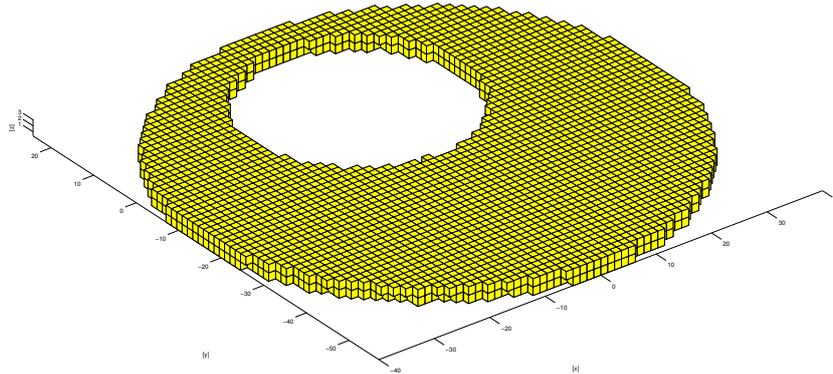


FIGURE 2.26 – Tube voxellisé (grossièrement)

ACP

D'après ce qui précède, la matrice X a la particularité de posséder beaucoup plus de colonnes que de lignes ($p \gg n$). Il est donc infiniment plus intéressant de diagonaliser XX^t . En conséquence, on cherche les vecteurs propres v_α de la matrice XX^t associés aux valeurs propres λ_α . Enfin, on reconstruit notre tableau de données X à partir des q premiers axes factoriels en utilisant la formule (2.31). Les coordonnées des points du nuage sur l'axe α dans \mathbb{R}^p sont par construction les composantes de Xu_α .

$$\tilde{X} = \sum_{\alpha=1}^q Xu_\alpha u_\alpha^t \quad \text{avec } u_\alpha = \frac{1}{\sqrt{\lambda_\alpha}} X^t v_\alpha \tag{2.31}$$

On applique notre **ACP** sur des géométries réparties sur l'ensemble de l'espace de conception. Il ne ressort pas de valeurs propres fortes comparées aux autres (voir le tableau 2.3) et il n'est pas possible de démarquer clairement des valeurs propres qui seraient prédominantes sur les autres. Autrement dit, il n'y a pas de réduction de dimension évidente au regard du critère de qualité globale (cf. équation (2.26)). Les valeurs propres ne permettent pas de résumer l'information. Pourtant, la représentation des composantes de Xu_α trois à trois donne une nappe dans l'espace, ce qui corrobore le fait que la vraie dimension du problème est de deux.

Dans un second temps, on refait le même exercice mais cette fois localement sur un sous-échantillonnage d'une centaine de géométries autour d'une position choisie. Comme la plage de variation des paramètres CAO est faible, beaucoup de relations linéaires sont introduites. De nouveau, il n'est pas possible de distinguer de valeurs propres fortes. On trouve que, localement, la dimension du problème est de 1 (voir la figure 2.28). La seule analyse des valeurs propres et du taux τ_q ne permet pas de conclure.

Contrairement à ce qu'affirmaient les auteurs de [Raghavan 2012, Raghavan 2013], il n'est pas possible de ne retenir que les modes les plus énergétiques. Du fait que la diagonalisation

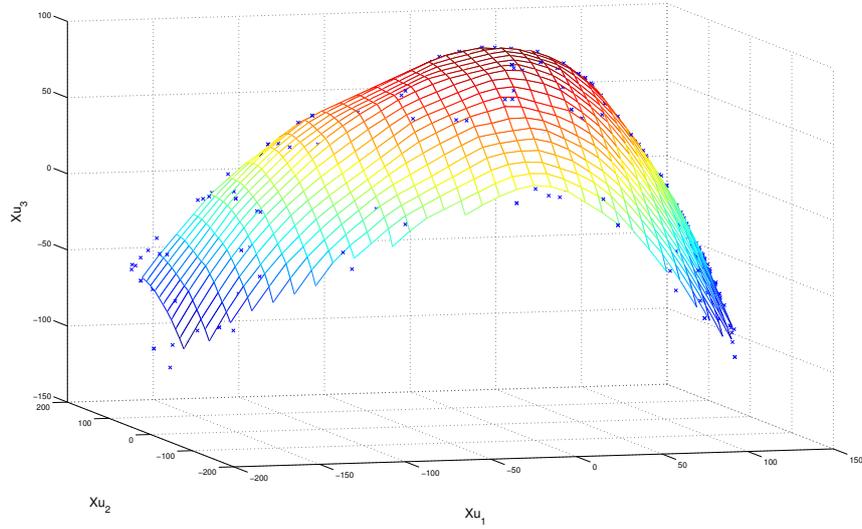


FIGURE 2.27 – Représentation des trois premières composantes principales

N°	Valeur propre	Qualité globale de reconstruction
	λ_α	τ_q
1	12277	0.43284
2	3979.9	0.57316
3	2899.4	0.67538
4	1376.1	0.7239
5	1066.8	0.76151
6	695.92	0.78604
7	575.37	0.80633
8	424.46	0.82129
9	357.55	0.8339
10	319.85	0.84518
11	292.22	0.85548
12	240.01	0.86394
13	203.25	0.87111
14	167.85	0.87703
15	161.8	0.88273
16	157.94	0.8883
17	140.34	0.89325
18	123.13	0.89759
19	115.55	0.90166
20	101.47	0.90524

Tableau 2.3 – Les 20 premières valeurs propres et la qualité globale de reconstruction du tube

supprime les redondances linéaires, l'ACP nous assure qu'on obtient une base de composantes indépendantes linéairement. Il faudrait donc appliquer une deuxième réduction sur ces compo-

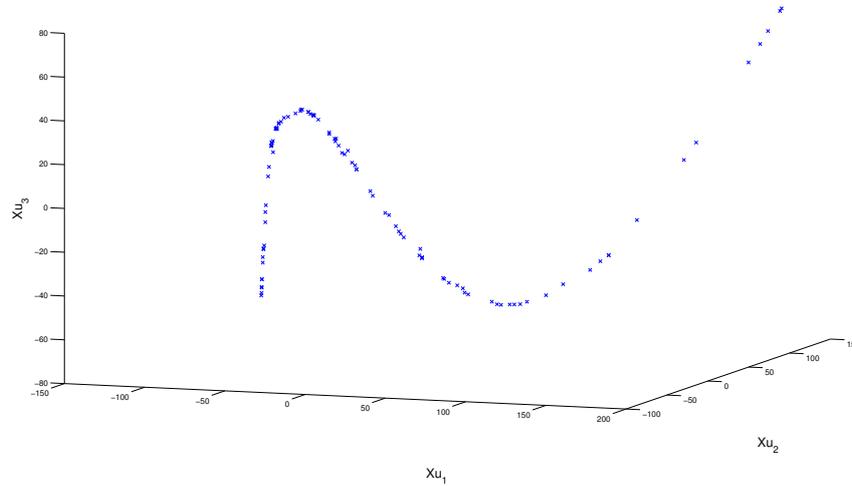


FIGURE 2.28 – Représentation des trois premières composantes principales autour d’une position

santes afin d’obtenir la variété. En résumé, on est seulement capable de déterminer la vraie dimension quand elle n’excède pas 2 par le biais d’un affichage graphique. Dans le cas contraire, on n’a pas de critère de décision qui nous permette de trancher.

2.3.1.4 Autres techniques

Il existe d’autres techniques non linéaires qui permettraient éventuellement de réduire la dimension de notre problème. Quand bien même nous réussirions à identifier la vraie dimension du problème, il faudrait ensuite :

- chercher l’optimum dans cet espace « courbe » ou « non linéaire » ;
- revenir ensuite à nos paramètres CAO de départ.

En se contentant uniquement à ce stade de réduire la dimension de notre matrice binaire, nous avons appliqué d’autres approches. On a utilisé une toolbox Matlab[®] (Matlab[®] Toolbox for Dimensionality Reduction [van der Maaten 2009]) développée par l’Université Technologique de Delft. Les techniques locales ou globales non linéaires qui ont, entre autres, été testées :

- la MDS (pour multidimensional scaling) : le positionnement multidimensionnel est un ensemble de techniques non linéaires qui mappent la représentation de données sur une représentation de plus petite dimension ;
- la Kernel PCA (pour principal component analysis) : l’analyse en composantes principales avec noyau est une reformulation de l’ACP qui permet d’exploiter des relations potentiellement non linéaires entre les variables. Le principe de la version non linéaire consiste à envoyer préalablement les données X par une application $\phi : X \rightarrow H$ dans un espace linéaire de grande dimension H muni d’un produit scalaire. C’est une approche utile pour la visualisation et la réduction du bruit [Scholz 2007] ;
- l’Isomap et la LLE (pour local linear embedding) : ces techniques ont été développées pour visualiser des données en grande dimension en les projetant dans un espace de plus

petite dimension mais la fonction de mapping n'est pas explicitement donnée.

Nous n'avons finalement pas prolongé le travail d'investigation plus loin car il nous est dès lors apparu un peu trop incertain et risqué. Nous avons préféré suivre une approche plus intéressante pour notre problème.

2.3.2 2^e voie : réseau de neurones artificiels

On cherche à identifier des relations entre nos paramètres CAO afin d'en réduire le nombre. On utilise pour cela un réseau de neurones artificiels qui va nous permettre de créer des *méta-paramètres*.

Le réseau de neurones est utilisé pour ses facultés de sélection des variables. Il est composé d'une succession de couches dont le nombre est défini préalablement. Du fait de la structure auto-associative, la première couche (*couche d'entrée*) et la dernière (*couche de sortie*) comportent le même nombre de neurones et le réseau est symétrique. On peut voir notre réseau comme un ensemble de deux sous-réseaux (voir la figure 2.29).

Le premier sous-réseau vise à réduire le nombre de paramètres représentatifs du jeu de données. On veut ainsi passer de p degrés de liberté (les paramètres CAO) à m degrés de libertés (les méta-paramètres) avec $m \ll p$. Pour ce faire, l'idée consiste à créer m combinaisons de paramètres initiaux. Ces combinaisons étaient linéaires en ACP, elles peuvent ici être plus complexes. Le premier sous-réseau est le résultat d'un multiplexage, le second est *a contrario* un démultiplexage [Vyshinsky 2010]. L'idée est de reconstituer un modèle des paramètres initiaux à partir des paramètres réduits. Cette fonction est essentielle, nous l'avons déjà évoquée dans le paragraphe précédent. Le degré de complexité des combinaisons est infini, il « suffit » d'augmenter le nombre de couches de neurones.

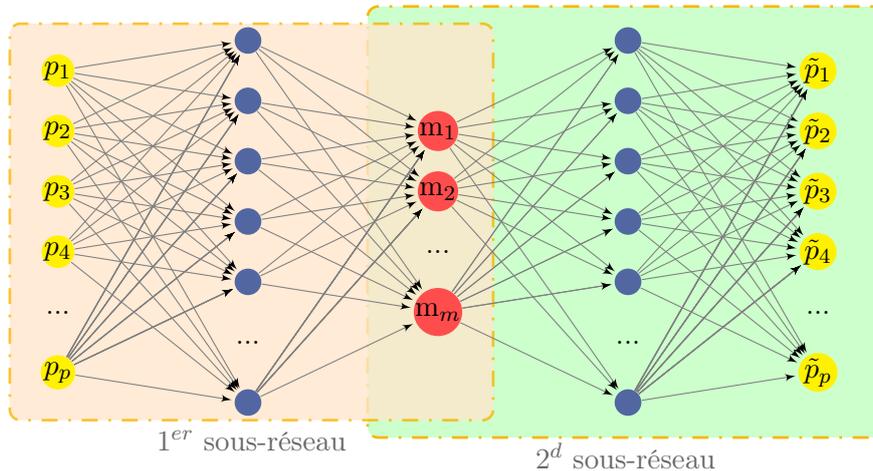


FIGURE 2.29 – Détail des sous-réseaux

Chacun des neurones des couches d'entrée et de sortie correspond à un paramètre CAO noté respectivement p_i et \tilde{p}_i . Entre ces deux couches se trouvent les *couches cachées*, dont la *couche centrale* qui est constituée de neurones formant les méta-paramètres, noté m_i . Chaque couche est composée de n_i neurones prenant leurs entrées sur les n_{i-1} neurones de la couche précédente (voir la figure 2.30).

L'utilisation de 3 couches cachées nous permet d'effectuer des ACP non linéaires [Hayes 2006], alors qu'un réseau à une seule couche cachée est limité aux ACP classiques [Bourlard 1988]. Le réseau force les sorties à être égales aux entrées correspondantes avec une grande précision. Pour cela, il est entraîné sur une base d'apprentissage, constituée dans notre cas par un jeu de géométries réalisables. Le problème consiste à trouver les matrices de poids IW et LW et les vecteurs de biais b^1 et b^2 optimaux qui minimisent l'erreur quadratique moyenne suivante :

$$E = \sum_{i=1}^p \|\tilde{p}_i - p_i\|^2 \tag{2.32}$$

Une fois que le réseau est entraîné, chaque sous-réseau est séparé. La première partie relie nos paramètres CAO à un nombre restreint de méta-paramètres qui peuvent être en quelque sorte assimilés à des *composantes principales* [Scholz 2012].

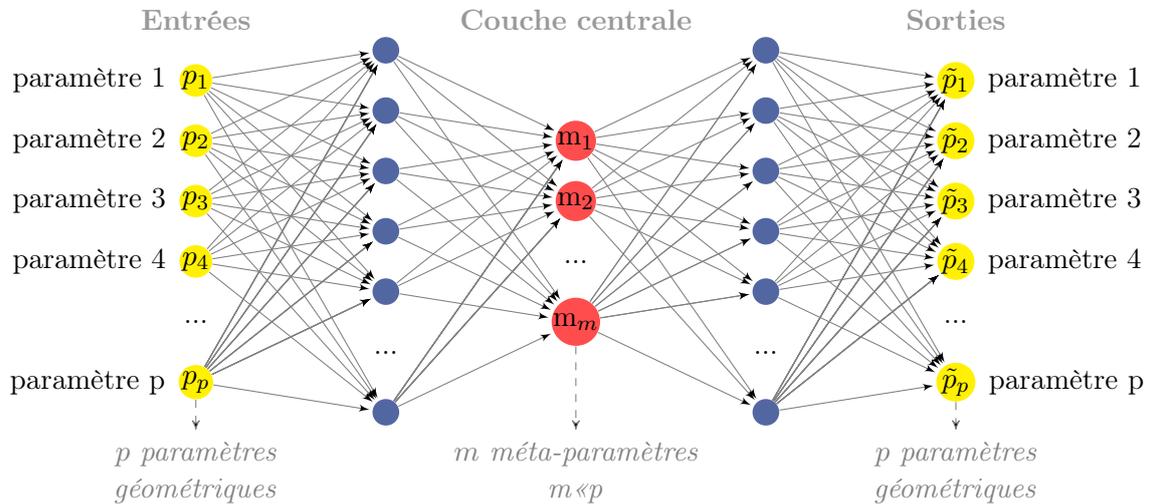


FIGURE 2.30 – Réseau de neurones artificiels auto-associatif

Chaque méta-paramètre représente une combinaison de paramètres géométriques dont l'influence est similaire vis-à-vis des simulations physiques. Un nombre désiré de méta-paramètres est fixé arbitrairement. Il dépend notamment du temps alloué pour l'étude. Ce nombre est ensuite ajusté par excès ou par défaut afin de minimiser les erreurs commises. L'espace de conception décrit par les paramètres géométriques est donc réduit intelligemment en parcourant un sous-espace intéressant pour atteindre rapidement la proximité de l'optimum.

Les méta-paramètres de la figure 2.31 sont obtenus par la formule suivante :

$$a_2 = f_2(LW f_1(IW p + b_1) + b_2) \tag{2.33}$$

avec :

- p : vecteur des entrées
- IW et LW : matrices des poids
- b^1 et b^2 : vecteurs des biais
- f_1 et f_2 : fonctions de transfert

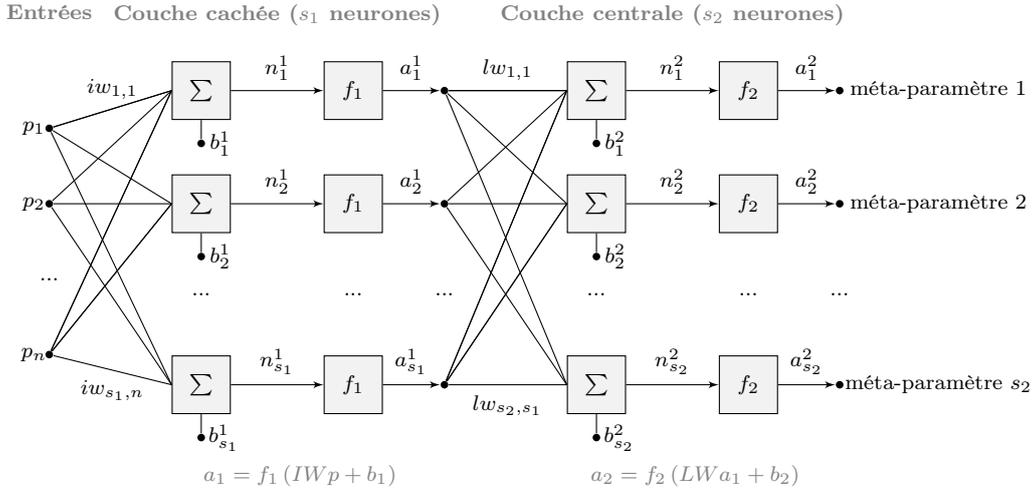


FIGURE 2.31 – Version symbolique du 1^{er} sous-réseau

Application

Nous avons appliqué la méthode à l'exemple du tube précédent. Ainsi, nous avons utilisé un réseau de type 5-6-2-6-5, c'est-à-dire avec 5 neurones dans les couches d'entrée et de sortie, 6 dans les couches cachées et 2 dans la couche centrale (2 méta-paramètres). Il n'existe pas de règle pour définir le nombre de neurones des couches cachées. Nous prenons en pratique un nombre de neurones supérieur au nombre de neurones d'entrée. Ce nombre est ajusté afin d'obtenir la meilleure performance du réseau en terme d'erreur quadratique moyenne. Les fonctions de transfert de la 2^e (f_1) et 4^e couche (f_3) sont des sigmoïdes, celle de la couche centrale (f_2) est linéaire.

Le réseau est entraîné sur un demi-millier de géométrie. L'erreur relative maximale entre les paramètres d'entrée et de sortie est de 8 %. Un des problèmes de cette méthode est de trouver le bon nombre de neurones empiriquement. Ce type de réseau est utilisé par la suite dans un cas industriel (partie 2.5 (p. 70)).

2.4 Approche multi-niveaux de fidélité

Une difficulté avec laquelle il faut composer en CFD est le coût calcul. Nous proposons ici une méthode pour contourner ce problème. Le phénomène est temporel mais les analyses nécessitent de lisser les informations sur le temps. Il existe deux façons de faire :

- soit on moyenne les équations dans un premier temps et on les résout ensuite ;
- soit on résout les équations dans un premier temps puis on moyenne les résultats.

Le premier point repose sur la *décomposition de Reynolds*, *i.e.* la décomposition d'une variable entre sa moyenne et ses fluctuations. Les équations instationnaires de Navier-Stokes sont moyennées afin de faire disparaître les termes de fluctuation rapides (qui sont de moyenne nulle). Ces équations moyennées dans le temps – connues sous le terme anglais *RANS equations* ou *Reynolds-averaged Navier-Stokes equations* – sont ensuite résolues numériquement. Ces simulations sont adaptées aux écoulements possédant des couches limites attachées. Elles ne donnent

accès qu'à des résultats stationnaires qui s'éloignent parfois de la réalité mais les calculs sont rapides.

Le second point renvoie à des méthodes plus précises mais aussi plus coûteuses. Elles consistent à résoudre directement les équations instationnaires et les solutions sont ensuite moyennées dans le temps. Ci-dessous sont listées quelques approches basées sur les équations de Navier-Stokes :

- la **simulation numérique directe** (SND, ou DNS acronyme anglais de **direct numerical simulation**) résout les équations de Navier-Stokes numériquement sans modèle de turbulence. Toutes les structures tourbillonnaires potentiellement présentes sont explicitement calculées. Toutefois, la DNS n'est pas utilisée dans un cadre industriel car elle nécessite des ressources informatiques excessives ;
- la **simulation des grandes échelles** (SGE, **large eddy simulation** ou LES en anglais) consiste à calculer par résolution numérique directe les structures turbulentes de grande taille et à modéliser celles de petite taille. Cependant, cette méthode est mal adaptée à la simulation de la dynamique pariétale, car elle nécessite un effort de maillage souvent prohibitif près de la paroi. Elle requiert moins de ressources informatiques que l'approche précédente ;
- les méthodes hybrides cherchent à combiner les avantages particuliers de chacune des approches de base afin d'optimiser le ratio coût/précision. Par exemple, la DES (pour **detached eddy simulation**) est une technique hybride proposée par [Spalart 1997] dans laquelle les couches limites attachées sont résolues par approche RANS alors que les régions décollées sont traitées par LES.

Nous proposons de tirer partie de ces deux niveaux de modélisation pour optimiser nos formes. On utilise un *modèle stationnaire* de type RANS pour identifier les zones intéressantes et un *modèle instationnaire* de type DES pour affiner notre recherche. Il est également possible d'utiliser une méthode de Lattice-Boltzmann en lieu et place des simulations instationnaires (DES). Au lieu de résoudre les équations de Navier-Stokes, l'équation discrète de Boltzmann est résolue pour simuler l'écoulement d'un fluide newtonien avec des modèles de collision tels que Bhatnagar-Gross-Krook.

2.5 Application de la méthode à un cas industriel

Au regard de ce que nous avons présenté précédemment, nous proposons ici une méthode d'optimisation qui va intégrer les développements précédents. Elle s'appuie sur un modèle CAO paramétré. Ce modèle va nous permettre de générer des géométries et d'apprendre la frontière de l'espace de conception en utilisant la SVM. Cette dernière nous aide à créer une base de géométries admissibles beaucoup plus importante, sur laquelle on va chercher le meilleur plan D-optimal constitué de k géométries.

L'évaluation de la fonction objectif se fait au travers des k géométries précédentes à l'aide du modèle physique simplifié. Cela nous permet de découvrir notre surface de réponse rapidement au détriment de la précision. En parallèle, les géométries admissibles sont utilisées dans un réseau de neurones auto-associatif pour créer un ensemble limité de méta-paramètres. Ceux-ci représentent une combinaison de paramètres ayant une influence similaire par rapport à la variation de formes. Cette méthode permet de réduire l'espace de conception intelligemment et rapidement, dégageant un sous-espace intéressant proche de l'optimum. Les méta-paramètres permettent de contourner

les difficultés liées à l'optimisation en grande dimension du fait que leur nombre m est nettement inférieur au nombre p de paramètres géométriques. Ainsi, la convergence vers la valeur optimale est légèrement dégradée.

In fine, les méta-paramètres obtenus sont utilisés avec une méthode classique d'optimisation (plan d'expériences, krigeage, algorithme évolutionnaire, etc.). Les points intéressants sont évalués avec un modèle physique complet. En définitive, l'équipe de conception est à même de faire des propositions. La figure 2.32 résume la boucle d'optimisation.

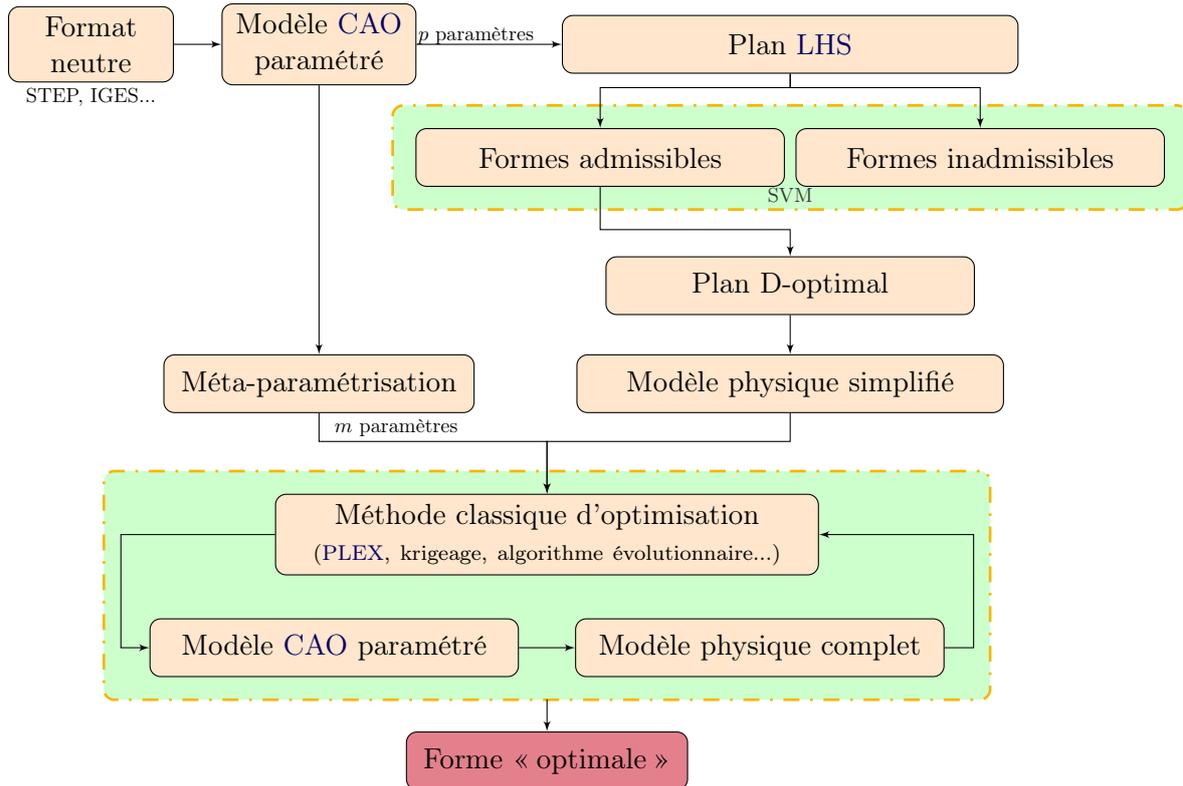


FIGURE 2.32 – Schéma-bloc de la méthode

2.5.1 Description du cas industriel

La boucle proposée est appliquée à un cas industriel. Nous nous concentrons sur l'optimisation de la forme extérieure d'un véhicule avec pour objectif de réduire son coefficient de trainée aérodynamique. Cela nécessite dans un premier temps de créer un modèle CAO paramétré qui respecte le thème design, c'est-à-dire qui conserve les lignes caractéristiques du véhicule. Ces lignes constituent les gènes d'un design automobile et se retrouvent notamment dans les lignes de carre et les lignes de lumière. Elles permettent de donner un tempérament au véhicule. Il est donc primordial pour l'ingénieur aérodynamicien de les respecter : un client achète avant tout un design avant d'acheter un véhicule aérodynamique. Ainsi, seules les modifications qui copient le thème design sont autorisées. Après une optimisation de la carrosserie, on doit être capable de reconnaître le design original sans aucune difficulté.

Il est important que le modèle CAO soit aussi stable que possible afin d'obtenir le meilleur taux de reconstruction possible. Par le passé, l'optimisation aérodynamique d'un véhicule se faisait essentiellement au moyen de séances en soufflerie. Ainsi, les paramètres physiques les plus influents au regard de la traînée aérodynamique sont connus. Des règles d'ingénierie ont été créées afin que les nouveaux modèles répondent à ces exigences dès les premières esquisses. Ces paramètres doivent donc être également des paramètres pilotant la CAO afin de corréliser les essais en soufflerie.

L'objectif est de réduire l'empreinte carbone des véhicules en minimisant le coefficient de traînée aérodynamique (C_x), c'est-à-dire en optimisant la pénétration dans l'air du véhicule. À grande vitesse, la résistance aérodynamique est prépondérante étant donné qu'elle évolue avec le carré de la vitesse. Elle est donc responsable en grande partie de la consommation du véhicule. Le C_x caractérise ainsi la performance aérodynamique du véhicule vis-à-vis de sa résistance à l'avancement dans l'air. Il est défini par l'équation (2.34). Dans la suite, nous nous intéresserons plus particulièrement au coefficient SC_x (cf. équation (2.35)) [Hucho 1998].

$$F_x = \frac{1}{2} \cdot \rho \cdot V^2 \cdot S \cdot C_x \quad (2.34)$$

$$SC_x = S \cdot C_x \quad (2.35)$$

avec :

- S : surface frontale (maître couple) ;
- ρ : masse volumique de l'air ;
- V : vitesse du véhicule.

Les essais en soufflerie sont réalisés à 160 km/h afin de limiter les effets de la viscosité. L'air est notamment considéré comme incompressible (il est d'usage de tenir compte de sa compressibilité à partir d'un nombre de Mach supérieur à 0,3).

Enfin, notons que le temps imparti pour évaluer le potentiel aérodynamique d'un véhicule et proposer des améliorations est très court (moins de 3 semaines).

2.5.2 Construction du modèle CAO

2.5.2.1 Véhicule

Pour cette étude, nous avons reconstruit la superstructure d'un modèle « archi ». Ce type de modèle définit les grandes cotes d'un véhicule à venir (Twingo iii) et permet d'évaluer les performances attendues. Il est notamment très simplifié au niveau des entrées d'air qui sont fermées et du soubassement. La reconstruction surfacique que l'on a développée sur ce véhicule a servi de base à l'approche décrite dans la section 2.1 (p. 36). C'est pourquoi le modèle présenté ci-dessous n'intègre pas totalement les nombreuses améliorations apportées depuis. C'est pour cela que le taux de reconstruction est certes bon (de l'ordre de 70 %) mais bien loin de ce que l'on obtient maintenant.

L'ensemble des paramètres physiques liés à la traînée aérodynamique a été intégré. Dans un second temps, d'autres paramètres sont venus enrichir la CAO afin d'assurer un maximum de

flexibilité et de balayer un espace de solutions conséquent. Nous avons décidé volontairement de limiter le nombre de paramètres à une vingtaine pour la suite de l'étude afin de ne pas monopoliser le serveur de calcul. Les plages de variation des paramètres sont définies selon les règles d'ingénierie. Au final, l'ensemble des paramètres est répertorié dans une table de paramétrage (angle du pare-brise, empattement, porte-à-faux arrière. . .) (cf. figure 2.33).

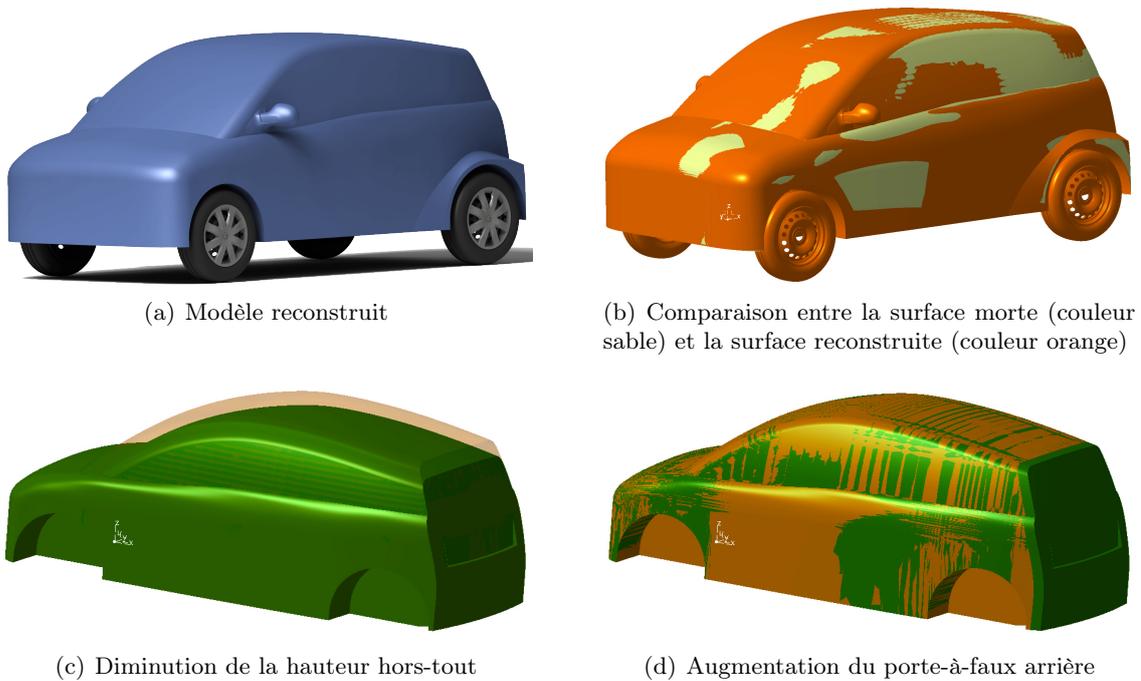


FIGURE 2.33 – Reconstruction surfacique d'un modèle « archi »

Au final, on teste le taux de reconstruction de notre modèle au moyen d'un plan LHS (ces plans sont détaillés dans l'annexe B.1.2 (p. 126)). Avec le logiciel CATIA® V5, cette opération prend environ 40 heures pour tester 14100 géométries en mode « batch » en monocœur. Le taux de reconstruction vaut 70,99 %. Ces informations vont nous permettre d'apprendre la frontière de notre espace de conception en utilisant une machine à vecteurs de support.

2.5.2.2 Veine de la soufflerie

Pour pouvoir confronter les résultats numériques aux résultats expérimentaux de la soufflerie S2A (située à Montigny-le-Bretonneux et issue du GIE entre PSA Peugeot Citroën, Renault et le CNAM), les écoulements de la veine vide de la soufflerie ont été mesurés afin de développer un modèle numérique capable de refléter au plus juste la géométrie réelle. Lorsque l'on passe un véhicule en soufflerie, il faut faire attention au facteur de blocage : la section la plus importante de la maquette (maître-couple) doit être inférieure à 5 % de la section totale de la veine de la soufflerie. En numérique, on n'est pas confronté à ce problème car les dimensions de la veine sont bien plus grandes que les dimensions réelles, cela afin d'éviter tout effet dû aux parois de la soufflerie. La veine numérique a une largeur de 27 m, une hauteur de 14 m et une longueur de 46 m (contre 6,5 m × 3,7 m × 14,7 m en réalité).

La soufflerie S2A possède un tapis roulant (6 m × 1 m) simulant le défilement de la route : cela est également reproduit dans la maquette numérique ainsi que la zone d'aspiration située devant le véhicule qui sert à limiter les décollements d'air qui peuvent avoir lieu. Enfin, les pneus sont coupés en partie afin de simuler leur écrasement.

2.5.3 Modélisation de la frontière du domaine de conception

N'ayant que peu de recul sur la méthode, nous avons étudié plus particulièrement trois paramètres : la longueur du pavillon, l'angle de la lunette et le porte-à-faux arrière. Leurs influences géométriques se font ressentir au niveau du bord de fuite du pavillon. On utilise un plan LHS pour générer 300 essais. La figure 2.34 montre qu'il existe des interactions entre ces paramètres. Lorsque qu'on applique une SVM sur ce plan en utilisant un noyau polynomial d'ordre 2, on discrimine à 100 %.

Au cours de l'étude de l'indépendance des variables, nous avons remarqué que certains échecs sont difficiles à expliquer. Ainsi, sur la figure 2.35(a), deux échecs liés au paramètre *galbe inférieur de la lunette* sont alignés sur une droite parallèle à un axe principal. Dans ce cas, il semble qu'il y ait une valeur du galbe inférieur de la lunette qui fasse échouer la génération des géométries. Pour s'en persuader, on a relancé un plan d'expériences en restreignant les bornes autour de la valeur suspecte, ce que confirme la figure 2.35(c). Il nous est très difficile d'expliquer ces échecs autrement que par un bogue du solveur CAO. Il y a donc un risque de classer une partie de l'espace en zone interdite et vice-versa.

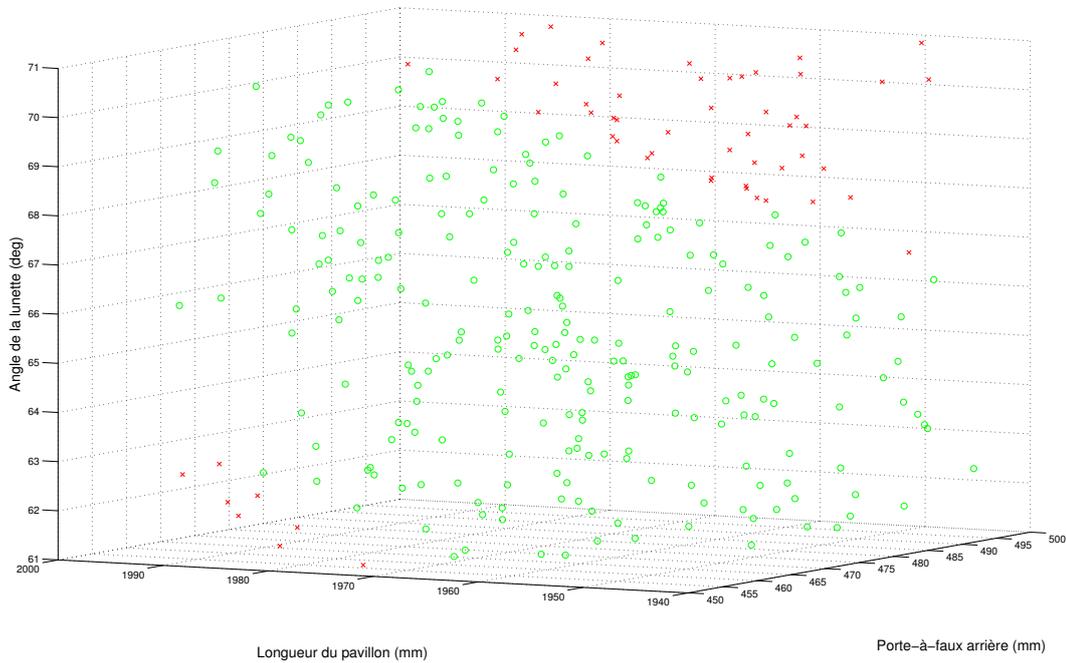
On souhaite maintenant modéliser notre domaine de conception pour notre modèle CAO complet, c'est-à-dire avec les 20 paramètres. Nous avons vu dans la partie 2.2.2 (p. 51) que l'on a plusieurs fonctions à notre disposition qui vont nous permettre de découper notre espace en différentes zones (d'un seul tenant ou non). On peut choisir d'utiliser :

- soit des fonctions simples comme des polynômes qui permettent d'obtenir des frontières régulières. S'ils accusent certaines limitations, il est tout à fait envisageable de passer à des techniques plus complexes ;
- soit des fonctions plus complexes (comme les fonctions de base radiale).

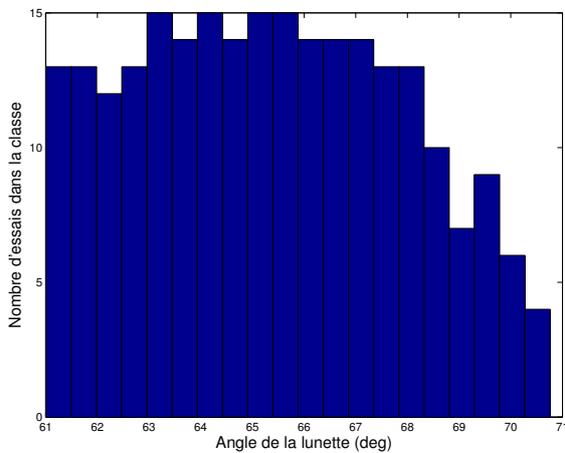
Nous avons utilisé les données précédentes (plan LHS de 14100 essais, taux de reconstruction de 70,99 %). Bien évidemment, à cause de la dimension du problème, une représentation graphique n'a pas de sens et nous ne pouvons que nous rattacher aux valeurs numériques pour estimer la qualité de notre discrimination. On obtient :

- soit un taux de discrimination de 90 % avec l'utilisation de polynômes d'ordre 2. L'algorithme ne permet pas de mieux discriminer certainement à cause du fait qu'il existe des échecs que l'on a du mal à expliquer comme certains bogues ;
- soit un taux de 98 % en utilisant un noyau gaussien. Non seulement le calcul est plus lourd, mais cela a surtout l'inconvénient de créer de nombreux îlots car l'algorithme apprend le bruit issu des bogues en quelque sorte.

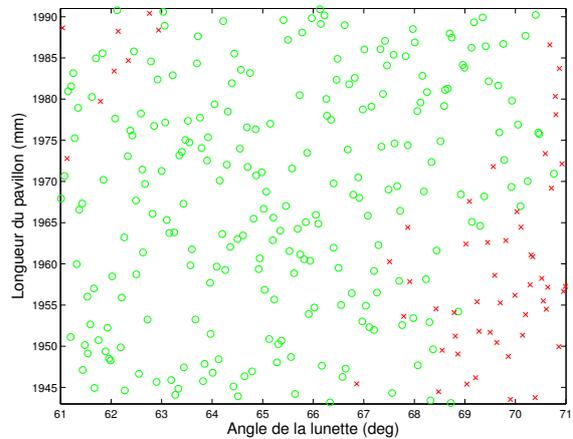
On peut décider de passer à un modèle plus complexe lorsque le taux de discrimination descend en dessous d'un seuil fixé (85 % par exemple). Il est également possible d'affiner la frontière en ajoutant des essais par la suite (en utilisant les plans auto-adaptatifs par exemple (voir l'annexe B.2.4 (p. 130))).



(a) Répartition des 300 essais dans l'espace (succès \circ /échecs \times)



(b) Histogramme des essais réussis en fonction de l'angle de la lunette

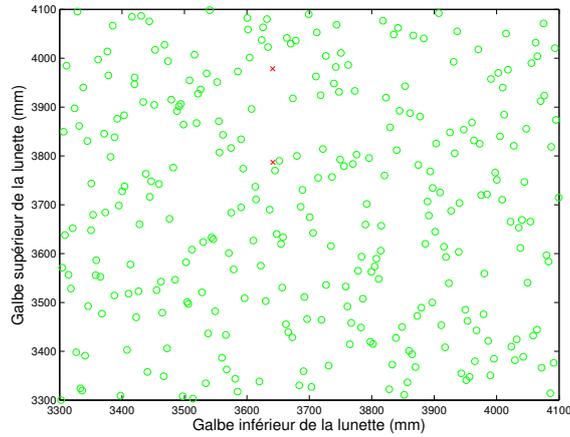


(c) Répartition des 300 succès \circ /échecs \times

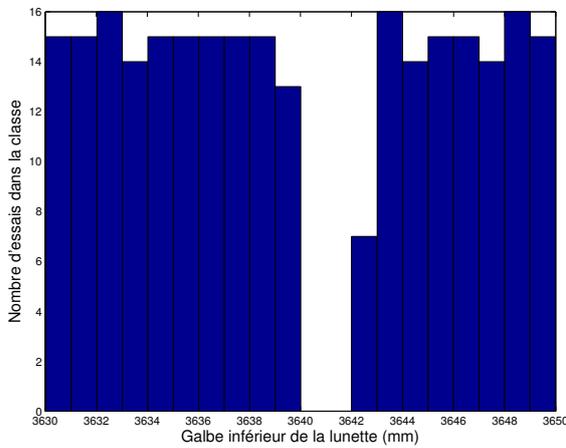
FIGURE 2.34 – Répartition des essais pour le triplet de paramètres (longueur du pavillon, angle de la lunette et porte-à-faux arrière)

2.5.4 Création des méta-paramètres

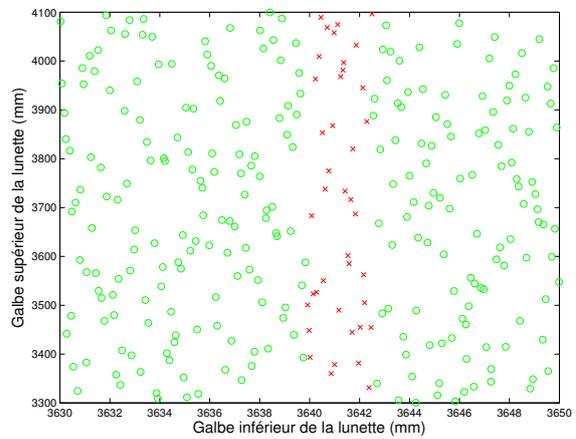
Les géométries classifiées bonnes sont utilisées dans un réseau de neurones artificiels équivalent à celui présenté dans la partie 2.3.2 (p. 67). Le nombre de méta-paramètres a été fixé à 10 afin de minimiser l'erreur quadratique moyenne, bien que nous visions 5 méta-paramètres au départ. La structure finale du réseau est de type 20-25-10-25-20. On observe une erreur relative maximale



(a) Répartition des succès \circ /échecs \times



(b) Histogramme des essais réussis en fonction de la valeur du galbe inférieur de la lunette



(c) Répartition des succès \circ /échecs \times

FIGURE 2.35 – Étude d’un paramètre particulier (*galbe inférieur de la lunette*) pour mettre en lumière un bogue du système CAO

entre les paramètres d’entrée et de sortie de l’ordre de 7 %.

2.5.5 Génération du plan d’expériences

A cause du fait qu’il existe des zones interdites sur le domaine de conception, la génération d’un plan d’expériences classique se déroule mal (taux d’échecs de 62,50 % notamment car les essais sont essentiellement situés sur les bords). On utilise donc le fait d’avoir appris la frontière. La SVM nous permet de classifier énormément d’essais et par conséquent de disposer d’une base de données conséquente pour générer le plan d’expériences qui va servir à construire notre surface de réponse. On suppose que notre étude suit un modèle polynomial du deuxième degré :

$$y = \beta_0 + \sum_{i=1}^p \beta_i x_i + \sum_{i=1}^p \beta_{ii} x_i^2 + \sum_{i < j}^p \beta_{ij} x_i x_j \quad (2.36)$$

Au départ, notre modèle comporte 20 paramètres ($p = 20$). En imaginant que toutes les interactions existent, il faudrait lancer autant de simulations que de coefficients à estimer, *i.e.* 231 (voir l'annexe B.3.1 (p. 135)). Nous avons réduit notre problème à une dizaine de méta-paramètres. Il nous faut donc 66 simulations en supposant que toutes les interactions existent. Nous avons utilisé un plan D-optimal de 70 essais. Comme notre classification n'est pas parfaite, tous les points ne sont pas faisables. Ces derniers ont été remplacés par les points possibles les plus proches.

2.5.6 Simulations

2.5.6.1 Création des fichiers

Dans ce paragraphe, nous nous attardons un peu sur la mise en place du modèle de simulation. On définit tout d'abord un template qui comporte la géométrie de la soufflerie, les valeurs des paramètres de maillage associés et le modèle physique stationnaire. La première étape consiste à préparer les fichiers qui seront envoyés au serveur en enrichissant le template :

- importation des géométries du véhicule et des rétroviseurs ;
- assemblage du véhicule et des rétroviseurs ;
- réparation de la géométrie ;
- recherche des lignes caractéristiques en vue du maillage ;
- réglage des paramètres du maillage pour le véhicule ;
- préparation des post-traitements.

Les fichiers précédents sont ensuite envoyés au serveur de calcul qui va dans les grandes lignes :

- générer le maillage ;
- supprimer les cellules non valides et améliorer le maillage ;
- lancer le calcul stationnaire ;
- calculer le maître-couple ;
- post-traiter les résultats stationnaires ;
- dans le cas où l'on souhaite un calcul instationnaire :
 - définir le modèle instationnaire ;
 - initialiser le calcul à partir de la solution stationnaire précédente ;
 - lancer le calcul instationnaire ;
 - post-traiter les résultats instationnaires.

2.5.6.2 Maillage

Toutes les géométries gravitent dans un espace proche de la géométrie initiale. On prend donc comme référence la géométrie initiale pour définir les zones de raffinement du maillage. Ces zones servent à capter l'ensemble des décollements qui pourraient intervenir, notamment au niveau du capot, des rétroviseurs, du soubassement et de la partie arrière (cf. figure 2.36).

La taille du maillage est définie en fonction du retour d'expérience des ingénieurs aérodynamiciens. On utilise un mailleur de type *trimmer* utilisant majoritairement des mailles hexaédriques avec un remaillieur surfacique. Un raffinement utilisant 12 couches de prismes (d'une épaisseur

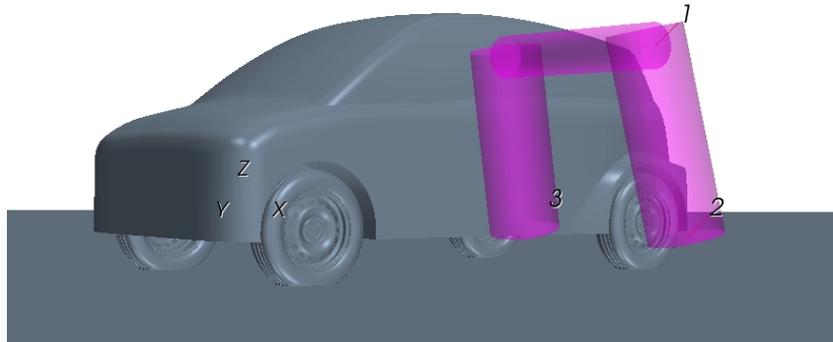


FIGURE 2.36 – Positionnement des zones de raffinement à l’arrière du véhicule

totale de 20 mm) au niveau des couches limites est utilisé afin de capturer les forts gradients de vitesse et de température. Le maillage final comporte 28 millions de cellules. Pour simuler la rotation des roues et le défilement du tapis roulant et des patins situés sous les roues, on définit une vitesse tangentielle sur la frontière de ces éléments équivalente à la vitesse du vent (160 km/h).

2.5.6.3 Modèle physique

Il est nécessaire de capter des différences de sensibilité assez faibles, de l’ordre de quelques millièmes de m^2 pour le SC_x . Il faut donc que le code de calcul soit assez prédictif pour pouvoir valider les variations des grandeurs physiques. Pour découvrir notre surface de réponse, on utilise alors dans un premier temps un modèle stationnaire de type RANS sur 3000 itérations. Lorsque l’on souhaite affiner des résultats, on prend un modèle DES (instationnaire) que l’on simule sur deux secondes de temps physique [Blazek 2005]. Les deux approches utilisent un modèle de turbulence de type $k-\omega$ avec un *segregated solver*.

2.5.7 Gestion de la boucle de calcul

Afin d’exploiter au maximum les ressources informatiques, l’intégralité de la boucle a été automatisée au moyen de scripts. La gestion du modèle CAO et la génération des géométries s’est faite avec le logiciel CATIA® V5 en utilisant notamment des scripts VB. Le logiciel Star-CCM+® (CD-Adapco) nous a permis de gérer l’ensemble des simulations de la mise en données au post-traitement, en passant par le maillage. Les échanges se font par le biais de scripts Java. Le lancement et la coordination de l’exécution des programmes s’est faite avec Matlab® (Math-Works), tout comme le réseau de neurones et la SVM. Enfin, le lancement des calculs sur serveur s’est fait avec le logiciel ProActive Parallel Suite® (OW2 consortium) au moyen de scripts XML. Ce logiciel permet entre autres d’optimiser le nombre de calculs à lancer simultanément sur le serveur HPC en fonction de son taux d’occupation. Comme la scalabilité de nos simulations est pratiquement parfaite jusqu’à 256 cœurs, une demi-dizaine de simulations est lancée en parallèle sur 64 ou 128 cœurs. En instationnaire, une simulation sur 64 cœurs prend environ 8 heures contre 3 jours pour le modèle DES.

2.5.8 Résultats

L'étude se concentre sur le coefficient de traînée aérodynamique que l'on cherche à minimiser. On post-traite cette valeur pour chaque géométrie. On s'intéresse également aux visualisations de certaines valeurs comme le coefficient de pression C_p :

$$C_p = \frac{p - p_\infty}{\frac{1}{2} \cdot \rho \cdot V_\infty^2} \quad (2.37)$$

avec :

- p : pression au point de mesure ;
- p_∞ : pression en entrée de soufflerie ;
- ρ : masse volumique de l'air ;
- V_∞ : vitesse de l'air en entrée de soufflerie.

L'équation de Bernoulli pour un fluide parfait incompressible est :

$$p + \frac{1}{2} \rho v^2 + \rho g z = \text{constante} \quad (2.38)$$

où p représente une énergie de pression, $\frac{1}{2} \rho v^2$ l'énergie cinétique et $\rho g z$ l'énergie potentielle (nulle pour nous). Il vient :

$$p + \frac{1}{2} \rho v^2 = p_\infty + \frac{1}{2} \rho V_\infty^2 \quad (2.39)$$

d'où :

$$C_p = 1 - \left(\frac{v}{V_\infty} \right)^2 \quad (2.40)$$

Loin des perturbations ($v = V_\infty$), le coefficient de pression est nul alors qu'en un point d'arrêt ($v = 0$), il vaut 1 (cf. figure 2.37). C_p peut être négatif par endroit comme au niveau du culot. On regarde également comment sont distribuées les lignes de courant autour du véhicule (comme le montre la figure 2.38). Les simulations numériques permettent d'apporter des informations de vitesse et de pression en tout point du domaine fluide, permettant alors d'observer des phénomènes qui ne sont pas accessibles expérimentalement. Par exemple, la contribution à la traînée aérodynamique totale de chaque partie du véhicule peut être déterminée, ce qui n'est pas envisageable lors d'essais en soufflerie.

Nous avons suivi les règles internes de Renault pour les post-traitements : les résultats en stationnaire sont obtenus en faisant la moyenne des 300 dernières itérations et en instationnaire, ils sont moyennés sur la dernière seconde. La comparaison des résultats avec les données expérimentales donne une précision du code de l'ordre de 2 % en instationnaire, soit environ 0,010 m² pour le SC_x . Cet écart est acceptable sachant que la répétabilité d'un même essai en soufflerie est de l'ordre de $\pm 0,003$ m².

Enfin, nous avons utilisé un algorithme polyMARS (voir la partie 1.3.1.2 (p. 20)) qui permet de créer des cassures pour tenir compte de la non-linéarité des données. En outre, cet algorithme est déjà disponible dans le logiciel Alternova[®] (développé conjointement par Renault et EURODECISION).

Après 66 simulations stationnaires et 5 simulations instationnaires, les résultats montrent un gain d'environ 35 millièmes de m² pour le SC_x (voir le tableau 2.4).

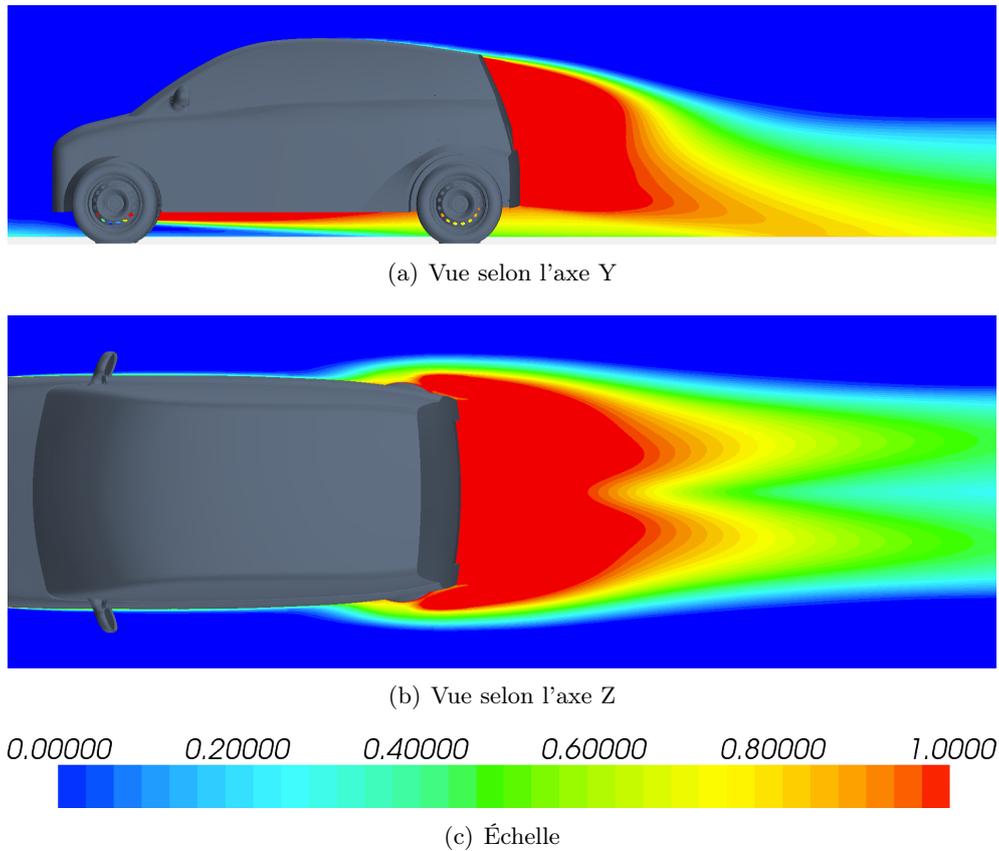


FIGURE 2.37 – Visualisation du coefficient de pression adimensionné

Configuration	S [m^2]	SC_x [m^2]
initiale	2.098	0.71498
optimale	2.088	0.68128

Tableau 2.4 – Valeurs du SC_x

2.6 Discussion de l'application

Notre étude s'est focalisée sur l'optimisation de la traînée aérodynamique uniquement, bien qu'il soit habituel de considérer d'autres sorties comme des valeurs locales du SC_x ou encore le coefficient de portance. La méthode que l'on propose s'adapte tout à fait à une fonction multi-objectif comme on le souligne dans l'article [Froment 2012]. Toutefois, ce n'est pas l'objet de la thèse.

Le gain de SC_x de l'ordre de 5 % peut paraître faible, cependant il représente un résultat intéressant pour un ingénieur aérodynamicien. Rappelons juste qu'un gain de 3 % correspond globalement à une diminution de 1 g de CO_2 par kilomètre. La traînée aérodynamique totale est due à l'écoulement externe autour du véhicule et des éléments de la carrosserie (miroirs, antenne, etc.) mais aussi à l'écoulement interne (refroidissement moteur, HVAC, freins, etc.). Il serait

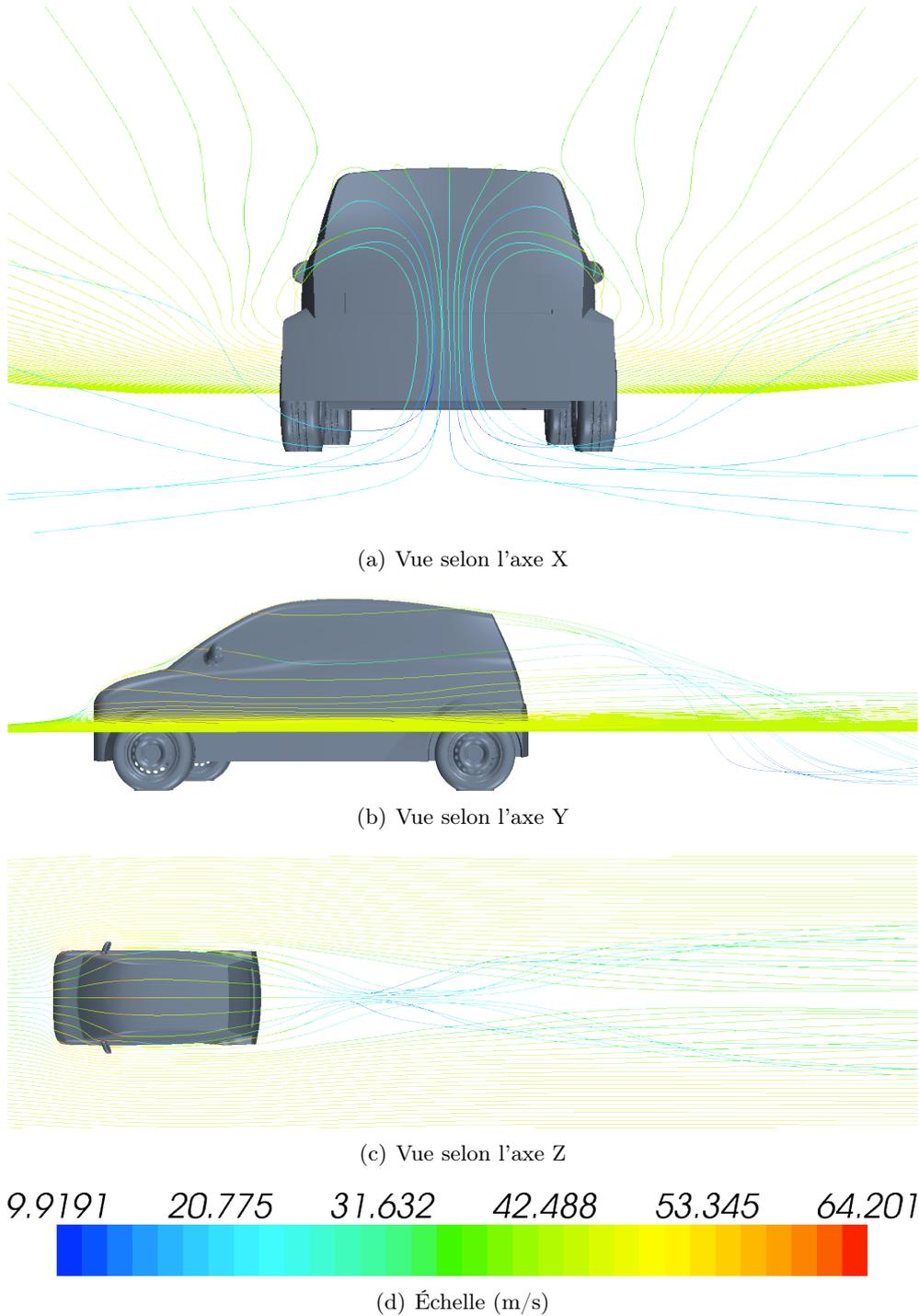


FIGURE 2.38 – Visualisation des lignes de courant (vitesse)

intéressant de mener la même étude en intégrant le compartiment moteur et le soubassement réel dans le modèle numérique.

La méthode proposée contourne les difficultés liées à une optimisation en grande dimension en

revenant dans une dimension acceptable pour appliquer des méthodes d'optimisation classiques (krigeage, etc.). Cela est réalisé grâce aux méta-paramètres qui permettent de réduire intelligemment l'espace de conception et de parcourir un sous-espace intéressant proche de l'optimum. L'utilisation de deux niveaux de fidélité est une solution pour maîtriser les temps de simulation. Cette approche rend possible une étude aérodynamique en une quinzaine de jours en ajustant notamment le nombre de méta-paramètres au temps alloué. Ce qui fait la force de la méthode est aussi son tendon d'Achille : il n'est pas possible de réduire outre mesure le nombre de paramètres. Il est nécessaire de chercher empiriquement le nombre de neurones à utiliser pour minimiser au mieux l'erreur quadratique moyenne. Certains paramètres peuvent avoir une faible influence géométrique, mais une forte influence sur la fonction objectif. Par exemple, la hauteur de la lame souple à l'avant du véhicule a une faible plage de variation, mais une très forte influence sur la traînée aérodynamique. Or, l'erreur commise par le réseau de neurones ne nous permet pas de maîtriser raisonnablement son comportement.

Il serait intéressant de personnaliser l'algorithme `polyMARS` afin de le dédier à l'optimisation en aérodynamique externe. On sait par exemple qu'il existe des valeurs limites pour l'angle de la lunette arrière entre lesquelles il y a de fortes variations. L'objectif serait d'intégrer dans l'algorithme des fonctions qui permettent de modéliser ces ruptures.

D'autre part, les `SVM` donnent de bons résultats quant à la recherche de l'espace de conception. L'application de cette théorie sur d'autres cas s'est faite avec succès. Elle sera intégrée dans le logiciel `Alternova`[®]. La méthode `CAO` donne des résultats extrêmement encourageants et les travaux sont poursuivis et appliqués à d'autres disciplines (notamment sur le retour `CAO` dans le projet `FUI RODIN`[®]).

En conclusion, les nombreuses techniques testées – dont le réseau de neurones – pour réduire la dimension de notre problème ne nous donnent pas entière satisfaction. La réduction du nombre de paramètres *a priori* est loin d'être évidente. La partie suivante change par conséquent de cap. Désormais, on ne se soucie ni du nombre de paramètres `CAO` comme précédemment ni de la réduction de la dimension. Afin de mener à bien l'optimisation en un temps acceptable, on va exploiter la connaissance des gradients donnés par les solveurs adjoints.

Bibliographie du 2^e chapitre

- [Albertelli 2005] M. Albertelli. *Projet MAPAO, plan d'expériences auto-adaptatif*. Rapport technique, Renault SA, 2005. (cité pages 52, 53, 54 et 130)
- [Blazek 2005] J. Blazek. *Computational fluid dynamics : principles and applications*. Elsevier, 2^e édition, 2005. (cité page 78)
- [Boser 1992] B.E. Boser, I.M. Guyon et V.N. Vapnik. *A training algorithm for optimal margin classifiers*. 1992. (cité page 52)
- [Bourlard 1988] H. Bourlard et Y. Kamp. *Auto-association by multilayer perceptrons and singular value decomposition*. *Biological Cybernetics*, vol. 59, pages 291–294, 1988. (cité page 68)
- [Chatterjee 2000] A. Chatterjee. *An introduction to the proper orthogonal decomposition*. *Current Science*, vol. 78, no. 7, 2000. (cité page 59)
- [Collobert 2000] R. Collobert. *Support vector machines : théorie et application*. Rapport technique, IDIAP, 2000. (cité page 55)
- [Cristianini 2000] N. Cristianini et J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000. (cité page 52)
- [Dhatt 2007] G. Dhatt, G. Touzot et E. Lefrançois. *Méthode des éléments finis*. Hermes, 2007. (cité page 41)
- [Frey 1999] P.J. Frey et P.L. George. *Maillages : applications aux éléments finis*. Hermes, 1999. (cité page 42)
- [Froment 2012] P. Froment, F. Gillot, L. Jézéquel et Y. Tourbier. *CAD-based multi-objective shape optimization method with meta-parameters*. *Proceedings of the 10th world congress on computational mechanics*, 2012. (cité page 80)
- [Gardan 2002] Y. Gardan. *CAO : modélisation géométrique*. *Techniques de l'ingénieur*, no. h3750, 2002. (cité page 44)
- [George 1991] P.L. George. *Génération automatique de maillages : applications aux méthodes d'éléments finis*. Masson, 1991. (cité pages 41 et 42)
- [Gunn 1998] S.R. Gunn. *Support vector machines for classification and regression*, 1998. (cité page 52)
- [Hayes 2006] J.C. Hayes, P.G. Doctor, T.R. Heimbigner, C.W. Hubbard, L.J. Kangas, P.E. Keller, J.I. McIntyre, B.T. Schrom et R. Suarez. *Application of artificial neural network modeling to the analysis of the automated radioxenon sampler-analyser state of health sensors*. *Proceedings of 28th Seismic Research Review*, pages 793–800, 2006. (cité page 68)
- [Hucho 1998] W.H. Hucho. *Aerodynamics of road vehicles*. SAE International, 4^e édition, 1998. (cité page 72)
- [Lebart 1982] L. Lebart, A. Morineau et J.-P. Fénélon. *Traitement des données statistiques*. Dunod, 2^e édition, 1982. (cité page 60)
- [Raghavan 2012] B. Raghavan. *Towards simultaneous meta-modeling for both the output and input spaces in the context of design shape optimization using asynchronous high-performance computing*. PhD thesis, Université de Technologie de Compiègne, 2012. (cité page 64)

Bibliographie du 2^e chapitre

- [Raghavan 2013] B. Raghavan, P. Breitkopf, Y. Tourbier et P. Villon. *Towards a space reduction approach for efficient structural shape optimization*. Structural and Multidisciplinary Optimization, vol. 48, no. 5, pages 987–1000, 2013. (cité pages 59 et 64)
- [Scholz 2007] M. Scholz, M. Fraunholz et J. Selbig. Principal manifolds for data visualization and dimension reduction, volume 58. Springer Berlin Heidelberg, 2007. (cité page 66)
- [Scholz 2012] M. Scholz. *Validation of nonlinear PCA*. Neural Processing Letters, vol. 36, no. 1, pages 21–30, 2012. (cité page 68)
- [Spalart 1997] P.R. Spalart, W.H. Jou, M. Stretlets et S.R. Allmaras. *Comments on the feasibility of LES for wings and on the hybrid RANS/LES approach*. Proceedings of the 1st AFOSR International Conference on DNS/LES, 1997. (cité page 70)
- [van der Maaten 2009] L.J.P. van der Maaten, E.O. Postma et H.J. van den Herik. *Dimensionality reduction : a comparative review*. Rapport technique, Tilburg University, 2009. (cité page 66)
- [Vyshinsky 2010] V.V. Vyshinsky, Y.A. Dorofeev et Y.N. Sviridenko. *Fast aerodynamic design technologies*. In 27th International Congress of the Aeronautical Sciences, 2010. (cité page 67)

Calcul de la sensibilité d'une prestation par rapport aux paramètres métier

Sommaire

3.1	État de l'art	86
3.2	Méthode proposée : morphing d'une forme à isoconnectivité	89
3.2.1	Paramétrisation d'une surface triangulée	90
3.2.2	Choix de la méthode	92
3.2.3	Transformations harmoniques	93
3.2.3.1	Cartes harmoniques discrètes avec des éléments finis linéaires	93
3.2.3.2	Unicité des cartes harmoniques avec l'hypothèse de Floater	94
3.2.3.3	Unicité des cartes harmoniques avec l'hypothèse de Remacle	95
3.3	Calcul des sensibilités en utilisant des transformations harmoniques 95	
3.3.1	Description théorique	95
3.3.2	Procédure de calcul de la sensibilité	96
3.3.3	Application sur un cas académique	97
3.4	Mise en place de la démarche	97
3.4.1	Présentation du cas d'étude	97
3.4.2	Choix du domaine paramétrique	98
3.4.3	Point invariant en 2D et sens de parcours de la frontière	100
3.4.4	Calcul du gradient au niveau des nœuds ou des éléments	101
3.4.4.1	Calcul des sensibilités au niveau des nœuds du maillage	102
3.4.4.2	Calcul des sensibilités au niveau des centroïdes des éléments	103
3.4.5	Valeurs des perturbations	103
3.5	Application de la méthode	104
3.5.1	Boucle d'optimisation	104
3.5.2	Cas d'étude	106
3.5.2.1	Cahier des charges en NVH	106
3.5.2.2	Cahier des charges en raideur	111
3.6	Discussion des résultats	115
	Bibliographie du 3^e chapitre	118

Lors de la conception d'un produit, les objectifs en termes de qualité, de coût, de délai et de prestations attendues – souvent contradictoires entre elles – conduisent à utiliser de façon intensive les méthodologies d'optimisation numériques. Le but est d'exploiter au mieux l'espace de conception disponible, tout en assurant un bon niveau de performance du produit.

Chapitre 3. Calcul de la sensibilité d'une prestation par rapport aux paramètres métier

Les processus de développement des pièces dans l'industrie reposent essentiellement sur des modèles CAO. C'est pour cela que les plans d'expériences y sont également populaires. Non seulement ces plans d'expériences ne nécessitent pas de connaissance minimale du phénomène étudié, mais surtout les paramètres CAO servent de paramètres d'optimisation. Comme on l'a vu précédemment, pour balayer l'espace de conception le plus vaste possible, il est nécessaire d'avoir une géométrie riche en degrés de liberté. Mais dans le même temps, il faut davantage de calculs pour trouver l'optimum.

Quand les gradients sont disponibles, ils peuvent largement nous aider à lever le verrou du nombre de degrés de liberté. Dans le chapitre précédent, on a vu comment réduire la dimension du problème. On va ici proposer une démarche basée sur les gradients qui ne nécessitent pas de faire cet exercice de réduction.

En mécanique des structures (dans le cadre des analyses linéaires), les gradients (par rapport au maillage, à la densité, etc.) existent depuis plusieurs années. Les éditeurs de code en mécanique des fluides (FLUENT[®] d'ANSYS, Star-CCM+[®] de CD-Adapco, Pamflow[®] et OpenFOAM[®] d'ESI...) tentent de fournir également des gradients depuis peu. On appelle *gradient* la sensibilité d'un critère (par exemple, une perte de charge) par rapport à des paramètres. En optimisation topologique, les paramètres sont la densité par élément ; en optimisation géométrique, ce sont la position des nœuds. Dans notre problème, les paramètres sont ceux de la CAO. La connaissance des sensibilités ouvre le champ des possibilités en optimisation : ils permettent d'envisager des techniques d'optimisation par gradient ou bien d'utiliser ces informations lors de l'exploitation des surfaces de réponse (co-krigeage, etc.) et de réduire ainsi le nombre de calculs nécessaires.

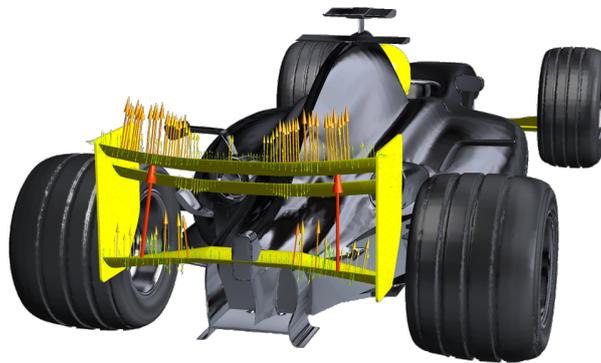


FIGURE 3.1 – Exemple de visualisation des sensibilités en mécanique des fluides. Ils indiquent comment corriger localement la forme pour améliorer le critère.

Dans ce chapitre, on désire développer une méthode d'optimisation qui utilise le gradient des prestations par rapport aux paramètres CAO quelle que soit la physique (structure, CFD...).

3.1 État de l'art

Avant d'aborder l'état de l'art, définissons tout d'abord quelques notations. On appellera par la suite J le critère (ou fonction objectif), α_i un paramètre CAO et par extension α l'ensemble des paramètres, Ω la forme géométrique associée aux paramètres précédents et $\tilde{\Omega}$ le maillage de

la forme Ω . Pour plus de clarté, les notations sont reprises schématiquement sur la figure 3.2. Soit $J = J(\alpha, \Omega(\alpha))$. On veut :

$$\begin{aligned} d_\alpha J &= \partial_\alpha J + \partial_\Omega J \times \partial_\alpha \Omega \\ &= \partial_\Omega J \times \partial_\alpha \Omega \end{aligned} \quad (3.1)$$

car on considère qu'il n'y a pas de dépendance directe entre le critère et nos variables d'optimisation, d'où le fait que le terme $\partial_\alpha J$ est nul.

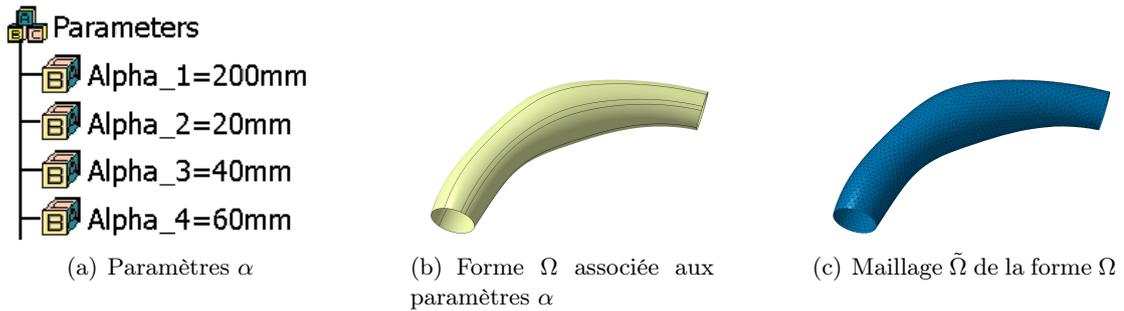


FIGURE 3.2 – Notations

On suppose que les éditeurs de code peuvent nous fournir $\partial_\Omega J$, ce qui peut être par exemple la sensibilité de J par rapport aux nœuds du maillage. Il faut calculer $\partial_\alpha \Omega$ ^a. Plusieurs solutions de principe permettent de calculer cette sensibilité :

- la voie *analytique* : on dérive les équations des surfaces CAO. Le principal avantage est que les résultats sont exacts à la précision du modelleur géométrique prêt [Yu 2011]. Cependant, cette méthode n'est envisageable que si la définition explicite de la géométrie est connue (*i.e.* la géométrie est vue comme une fonction dont on connaît l'expression mathématique formelle), ce qui n'est pas le cas avec la majeure partie des systèmes CAO qui sont des codes propriétaires ;
- la voie *numérique* sur les géométries CAO : l'évaluation du gradient est réalisée par différences finies comme le propose l'article [Armstrong 2007]. Ces méthodes se distinguent par le fait qu'elles comparent les modèles initial et perturbé avant le processus de maillage. Par exemple, [Chen 1997] explique que les sensibilités des sommets d'une surface sont calculées dans un premier temps, de ces sensibilités sont interpolées celles des nœuds situés sur le bord et enfin les gradients des nœuds de la surface sont évalués. De telles approches nécessitent que les représentations géométriques des courbes et des surfaces soient disponibles comme le mentionne l'article [Hardee 1999]. De surcroît, la perturbation d'un paramètre, même minime, entraîne parfois un changement de topologie : une même surface peut se retrouver composée d'un nombre de patches différents, ce qui amplifie les difficultés lors de l'identification de surface à surface [Robinson 2009]. D'autre part, l'identifiant d'un patch peut être amené à changer durant la génération de la forme perturbée [Kripac 1997] ;
- la voie *numérique* sur les maillages : on utilise aussi des différences finies mais cette fois entre les maillages de la forme initiale et de la forme perturbée. Cette approche, *a priori* plus simple, est loin d'être évidente. Il s'agit de déterminer le champ de déformation qui

a. La sensibilité $\partial_\alpha \Omega$ est couramment appelée *design velocity* dans la littérature anglo-saxonne.

va projeter un maillage sur l'autre tout en évitant de grosses distorsions de maille. De nombreuses méthodes s'appuient sur la déformation du maillage initial [Toivanen 2006] qui peut néanmoins introduire des distorsions. Comme les perturbations appliquées sont petites, de nombreuses techniques se basent sur un lissage Laplacien [Lindby 1999] : les nœuds sont projetés sur les surfaces géométriques de la nouvelle CAO durant l'opération de lissage. D'autres techniques sont discutées dans [Choi 1994, Choi 2005].

L'approche *analytique* est la plus propre mais elle n'est pas possible car les codes des systèmes CAO sont fermés. De même, le risque de changement de topologie (c'est-à-dire lorsque le nombre de patches change) rend également impossible la deuxième méthode basée sur la représentation de la géométrie CAO. Nous allons donc nous concentrer sur la troisième voie qui utilise les éléments finis.

Cette approche nécessite de passer par les formes discrétisées. On rappelle dans la figure 3.3 la façon naturelle de générer une forme perturbée à partir d'un modèle CAO.

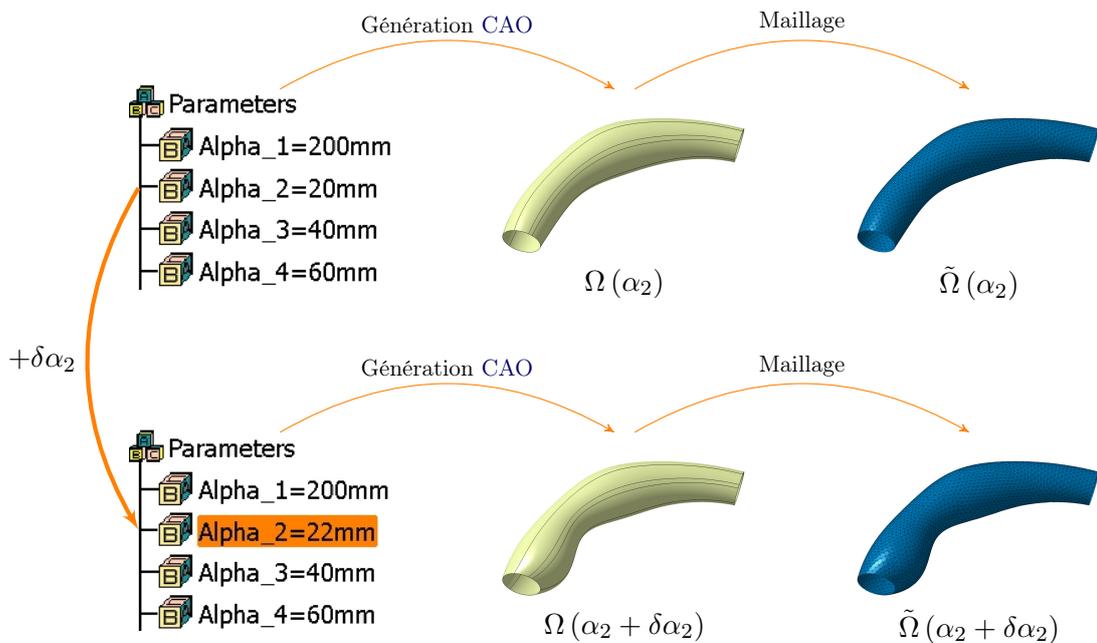


FIGURE 3.3 – Processus d'évaluation de la sensibilité d'une forme vis-à-vis du paramètre α_2

On voudrait évaluer la sensibilité de la forme par rapport à un paramètre α_i en utilisant l'équation (3.2). Pour cela, il faudrait que les deux maillages soient comparables, *i.e.* avoir une isoconnectivité entre les nœuds. Or, nous sommes en présence de deux maillages $\tilde{\Omega}(\alpha_i)$ et $\tilde{\Omega}(\alpha_i + \delta\alpha_i)$ non structurés et indépendants l'un de l'autre. D'où la question : comment construire $\tilde{\Omega}(\alpha_i + \delta\alpha_i)$?

$$\partial_{\alpha_i} \Omega \simeq \frac{\tilde{\Omega}(\alpha_i + \delta\alpha_i) - \tilde{\Omega}(\alpha_i)}{\delta\alpha_i} \quad i = 1 \dots p \quad (3.2)$$

1^{re} proposition : projection d'un maillage sur un autre

Cette idée consiste à projeter un maillage sur un autre selon la normale comme le propose [Robinson 2009]. Les formes nominale et perturbée sont maillées indépendamment l'une de l'autre, avec une taille de maille plus petite pour le maillage de la forme perturbée. Chaque nœud des éléments du maillage de la forme non perturbée est projeté selon la normale sur les éléments du maillage de la forme perturbée. Le nœud le plus proche sur le maillage de la forme perturbée est considéré comme le projeté du nœud pris sur le maillage initial (puisque les perturbations sont faibles). Le déplacement de chaque point dû à la perturbation peut être évalué comme la différence entre les coordonnées du point initial et celles de son projeté identifié. Cette méthode n'assure pas de trouver une projection à chaque fois, ce qui est mentionné par les mêmes auteurs dans l'article [Robinson 2012]. En cas d'échec, ils relancent la procédure en ajustant certains paramètres de la méthode.

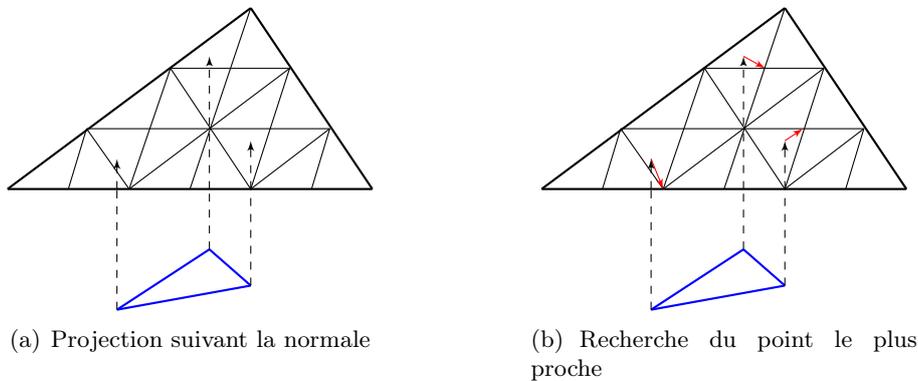


FIGURE 3.4 – Description schématique de la méthode proposée dans [Robinson 2012]

Toutefois, la projection pose parfois problème. Sur la figure 3.5(a), on dilate le rectangle bleu. Une partie de la forme dilatée ne rejoint aucun point du maillage initial. La projection a fonctionné mais le résultat est insuffisant car on n'a pas distribué les nœuds sur toute la forme. Sur la figure 3.5(b), on dilate en largeur le rectangle bleu tout en le contractant en hauteur : le coin supérieur droit glisse vers le bas. Un ensemble de nœuds ne trouve alors plus de projection. Si l'on ne sait pas gérer ces deux cas simples, tout le reste ne sera qu'une succession d'ajustages.

3.2 Méthode proposée : morphing d'une forme à isoconnectivité

Les approches que nous avons abordées précédemment montrent rapidement leurs limites. Nous proposons donc une nouvelle démarche qui morphe le maillage de la forme initiale sur le maillage de la forme perturbée tout en gardant l'isoconnectivité entre les deux maillages pour obtenir le déplacement de chaque nœud. Cette démarche se base sur le dépliage d'une géométrie 3D sur une forme canonique 2D. Cette idée peut être comparée au dépliage du globe terrestre sur un planisphère. Nous sommes partis du constat que, quelles que soient deux surfaces ayant la même topologie, il est possible de trouver une application bijective unique de l'une sur l'autre [Gu 2008].

Ces techniques dites de paramétrisation des surfaces sont couramment utilisées [Gu 2008] pour :

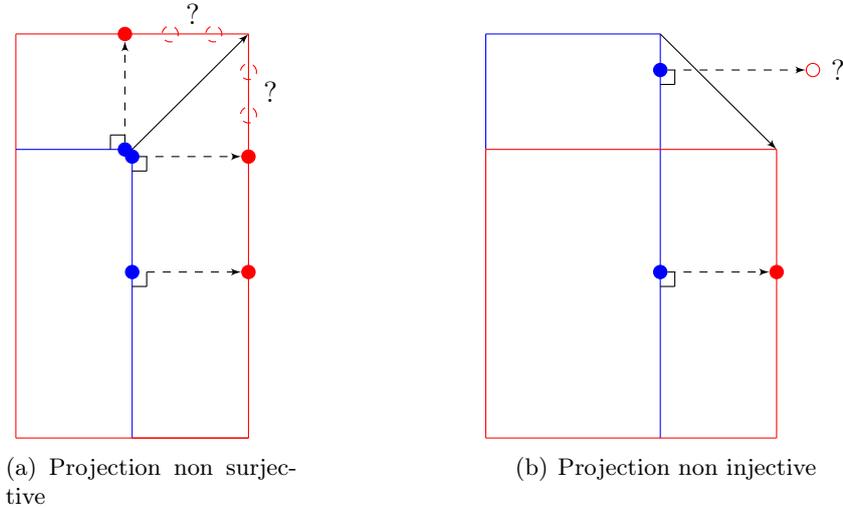


FIGURE 3.5 – Illustration des problèmes de la projection selon la normale

- améliorer la richesse visuelle des images générées par ordinateur en infographie ;
- remailler des surfaces 3D afin d'améliorer la qualité du maillage ;
- morpher une forme sur une autre ;
- réparer un maillage (par exemple, pour fermer un trou après une acquisition) ;
- etc.

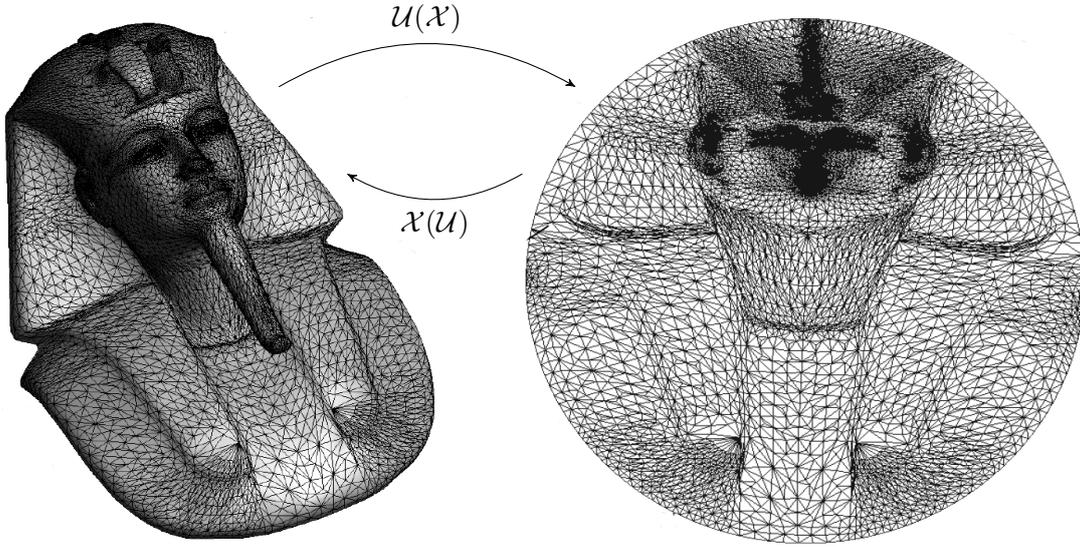
3.2.1 Paramétrisation d'une surface triangulée

Etant donné une surface \mathcal{S} de \mathbb{R}^3 , une *paramétrisation* $\mathcal{X}(\mathcal{U})$ de \mathcal{S} est une fonction bijective mettant un sous-ensemble \mathcal{S}' de \mathbb{R}^2 en correspondance avec la surface \mathcal{S} (voir figure 3.6). Elle admet une fonction inverse $\mathcal{U}(\mathcal{X})$ appelée *paramétrisation inverse* :

$$\mathcal{U} : \mathcal{X} \in \mathcal{S} \subset \mathbb{R}^3 \rightarrow \mathcal{U}(\mathcal{X}) \in \mathcal{S}' \subset \mathbb{R}^2 \tag{3.3}$$

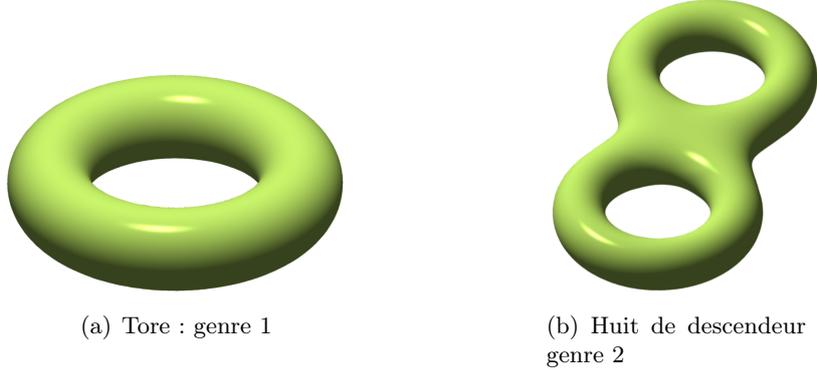
qui transforme continûment une surface \mathcal{S} de \mathbb{R}^3 en une surface \mathcal{S}' de \mathbb{R}^2 pourvu que les deux surfaces \mathcal{S} et \mathcal{S}' aient la même topologie, c'est-à-dire le même genre $G(\mathcal{S})$ et le même nombre de frontières N_b [Remacle 2010]. La surface canonique sur laquelle la forme est dépliée est usuellement appelée *domaine paramétrique*.

En dimension deux d'espace, une même topologie revient à dire que les deux surfaces ont le même nombre de « trous » (ou de composantes connexes de la frontière). En dimension trois, cette vision est plus compliquée car non seulement il faut prendre en compte les « trous » mais aussi les « anses » et les « boucles », *i.e.* le genre du domaine. On définit le genre d'une surface comme le nombre maximum de courbes fermées indépendantes. Autrement dit, dans le procédé de détermination du genre, le complément de la réunion de ces courbes reste connexe. Deux surfaces n'ayant pas le même genre ne sont pas homéomorphes, c'est-à-dire qu'il n'existe pas d'application bijective continue entre les deux espaces topologiques dont la réciproque est continue. Par exemple, une sphère et un disque sont de genre 0, un tore de genre 1 et un huit de descendeur de genre 2 (cf. figure 3.7).



(a) Masque de Toutânkhamon en 3D

(b) Masque de Toutânkhamon déplié sur un disque

FIGURE 3.6 – Illustration du dépliage du masque de Toutânkhamon sur un disque (image extraite de l'article [Remacle 2010])


(a) Tore : genre 1

(b) Huit de descendeur : genre 2

FIGURE 3.7 – Exemple du genre de deux surfaces

Dans ce qui suit, nous considérons que la surface \mathcal{S} est munie d'une triangulation $\mathcal{S}_{\mathcal{T}}$ ($\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_N\}$). Ce type de maillage est facile à obtenir à partir des systèmes CAO en exportant la géométrie en STL. Soit un maillage triangulaire comportant N_v sommets (v pour *vertices*), N_e arêtes (e pour *edges*) et N_t triangles. Le genre de la surface est donné par la formule d'Euler-Poincaré :

$$G(\mathcal{S}_{\mathcal{T}}) = \frac{-N_v + N_e - N_t + 2 - N_b}{2} \quad (3.4)$$

Chaque sommet $V_i|_{i=\llbracket 1, N_v \rrbracket}$ du maillage a donc des coordonnées dans l'espace $X_i = (x_i, y_i, z_i) \in \mathcal{S}$ et des coordonnées paramétriques $U_i = (u_i, v_i) \in \mathcal{S}'$. Isolons un triangle \mathcal{T}_j dont les sommets sont notés V_1, V_2 et V_3 . Cet élément peut être paramétré dans \mathcal{S} par une fonction classique d'interpolation linéaire des méthodes d'éléments finis :

$$\mathcal{X}(\xi) = (1 - \xi - \eta)X_1 + \xi X_2 + \eta X_3 \quad (3.5)$$

De manière analogue, on peut paramétrer ce triangle dans \mathcal{S}' :

$$\mathcal{U}(\xi) = (1 - \xi - \eta)U_1 + \xi U_2 + \eta U_3 \quad (3.6)$$

Afin de calculer la paramétrisation $\mathcal{X}(\mathcal{U})$, on inverse d'abord l'équation (3.6) :

$$\mathcal{U} - U_1 = \underbrace{\begin{pmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{pmatrix}}_{\mathcal{U},\xi} \underbrace{\begin{pmatrix} \xi \\ \eta \end{pmatrix}}_{\xi} \quad (3.7)$$

ce qui donne :

$$\xi(\mathcal{U}) = (\mathcal{U},\xi)^{-1}(\mathcal{U} - U_1) = (\xi,\mathcal{U})(\mathcal{U} - U_1) \quad (3.8)$$

avec :

$$\xi,\mathcal{U} = \frac{\begin{pmatrix} v_3 - v_1 & -(u_3 - u_1) \\ -(v_2 - v_1) & u_2 - u_1 \end{pmatrix}}{(u_2 - u_1)(v_3 - v_1) - (v_2 - v_1)(u_3 - u_1)} \quad (3.9)$$

La paramétrisation $\mathcal{X}(\mathcal{U})$ peut donc être calculée en trois étapes :

1. recherche du triangle \mathcal{T}_j du domaine paramétrique \mathcal{S}' qui contient le point \mathcal{U} ;
2. calcul des coordonnées locales $\xi = (\xi,\eta)$ du point \mathcal{U} dans le triangle \mathcal{T}_j en utilisant l'équation (3.8) ;
3. calcul de la paramétrisation $\mathcal{X}(\mathcal{U}) = \mathcal{X}(\xi(\mathcal{U}))$ en utilisant l'équation (3.5).

Les procédures de génération de maillage nécessitent non seulement de calculer la paramétrisation $\mathcal{X}(\mathcal{U})$ mais aussi ses dérivées \mathcal{X},\mathcal{U} . Nous avons :

$$\mathcal{X},\mathcal{U} = \mathcal{X},\xi \xi,\mathcal{U} = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{pmatrix} \frac{\begin{pmatrix} v_3 - v_1 & -(u_3 - u_1) \\ -(v_2 - v_1) & u_2 - u_1 \end{pmatrix}}{(u_2 - u_1)(v_3 - v_1) - (v_2 - v_1)(u_3 - u_1)} \quad (3.10)$$

Il est souvent utile de calculer le tenseur métrique fondamental $M = (\mathcal{X},\mathcal{U})^t \mathcal{X},\mathcal{U}$ qui caractérise comment une transformation $\mathcal{X}(\mathcal{U})$ altère les distances et les angles.

3.2.2 Choix de la méthode

Parmi les méthodes de paramétrisation linéaires, il y a notamment les transformations isométriques qui préservent les angles, les aires et les longueurs. Toutefois, elles n'existent que pour des cas particuliers. C'est pour cela que l'on se tourne vers des approches universelles basées sur la minimisation de l'énergie de Dirichlet :

- les transformations conformes : elles préservent les angles et n'imposent pas de conditions aux bords ;
- les cartes harmoniques : elles imposent des conditions aux bords.

On souhaite par la suite comparer les domaines paramétriques de deux formes : il est par conséquent nécessaire d'imposer les conditions aux bords. On se tourne donc vers les cartes harmoniques qui, bien qu'elles ne respectent pas les angles, possèdent d'autres propriétés intéressantes :

1. elles sont faciles à coder et nécessitent seulement la résolution d'un problème linéaire ;
2. elles sont indépendantes de la triangulation initiale ;
3. elles sont indéfiniment différentiables sur une surface ;
4. la solution est unique pour des régions convexes.

3.2.3 Transformations harmoniques

Nous cherchons à déplier une surface \mathcal{S} de l'espace sur un disque unité \mathcal{S}' . Par conséquent, la surface \mathcal{S} doit être de genre 0 avec au moins une frontière, notée $\partial\mathcal{S}_1$. On précise que le disque unité \mathcal{S}' devra contenir autant de trous que \mathcal{S} a de frontières moins une. Les coordonnées u et v sont calculées séparément comme solution des problèmes de Laplace qui suivent. On impose des conditions aux limites de type Dirichlet sur la frontière $\partial\mathcal{S}_1$ et de type Neumann pour les autres frontières éventuelles $\partial\mathcal{S}/\partial\mathcal{S}_1$.

$$\begin{cases} \nabla^2 u = 0, & \nabla^2 v = 0 & \text{sur } \mathcal{S}, \\ u = \bar{u}(\mathcal{X}), & v = \bar{v}(\mathcal{X}) & \text{sur } \partial\mathcal{S}_1, \\ \frac{\partial u}{\partial n} = 0, & \frac{\partial v}{\partial n} = 0 & \text{sur } \partial\mathcal{S}/\partial\mathcal{S}_1. \end{cases} \quad (3.11)$$

Nous devons ensuite trouver des fonctions $\bar{u}(\mathcal{X})$ et $\bar{v}(\mathcal{X})$ qui déplient la frontière $\partial\mathcal{S}_1$ sur le cercle unité. Pour cela, nous choisissons arbitrairement un sommet initial V_s (s pour *start*) et nous calculons la distance l_i de V_s à V_i le long de la frontière $\partial\mathcal{S}_1$. Si L est la longueur totale de $\partial\mathcal{S}_1$, les conditions limites suivantes :

$$\begin{cases} u(X_i) = \cos \frac{2\pi l_i}{L} \\ v(X_i) = \sin \frac{2\pi l_i}{L} \end{cases} \quad (3.12)$$

déplient $\partial\mathcal{S}_1$ sur le cercle unité.

Ce dépliage est unique dans sa version continue sous réserve que la surface \mathcal{S}' soit convexe [Remacle 2010]. La définition du domaine paramétrique (cercle, ellipse, rectangle, etc.) n'apporte pas de différences significatives.

3.2.3.1 Cartes harmoniques discrètes avec des éléments finis linéaires

Nous nous appuyons pour les formulations qui suivent sur la proposition de [Remacle 2010]. Le système (3.11) est équivalent au problème de minimisation quadratique qui suit :

$$\min_{\mathcal{U} \in U(\mathcal{S})} J(\mathcal{U}) = \frac{1}{2} \int_{\mathcal{S}} \|\nabla \mathcal{U}\|^2 ds \quad \text{avec } U(\mathcal{S}) = \{\mathcal{U} \in H^1(\mathcal{S}), \mathcal{U} = f(\mathcal{X}) \text{ sur } \partial\mathcal{S}\} \quad (3.13)$$

avec H^1 l'espace de Sobolev. En prenant \mathcal{U} comme :

$$\mathcal{U}_h(\mathcal{X}) = \sum_{i \in I} U_i \phi_i(\mathcal{X}) + \sum_{i \in J} f(X_i) \phi_i(\mathcal{X}) \quad (3.14)$$

où I représente l'ensemble des nœuds de la triangulation qui n'appartiennent pas à la frontière de Dirichlet et J l'ensemble des nœuds qui appartiennent à la frontière de Dirichlet. ϕ_i sont des fonctions de forme associées aux nœuds du maillage avec $\phi_i(X_j) = \delta_{ij}$. La fonctionnelle J de l'équation (3.13) devient avec l'équation (3.14) :

$$\begin{aligned} J(U_1, \dots, U_N) = & \frac{1}{2} \sum_{i \in I} \sum_{j \in I} U_i U_j \int_{\mathcal{S}_T} \nabla \phi_i(\mathcal{X}) \cdot \nabla \phi_j(\mathcal{X}) \, ds + \\ & \sum_{i \in I} \sum_{j \in J} U_i f(X_j) \int_{\mathcal{S}_T} \nabla \phi_i(\mathcal{X}) \cdot \nabla \phi_j(\mathcal{X}) \, ds + \\ & \frac{1}{2} \sum_{i \in J} \sum_{j \in J} f(X_i) f(X_j) \int_{\mathcal{S}_T} \nabla \phi_i(\mathcal{X}) \cdot \nabla \phi_j(\mathcal{X}) \, ds \end{aligned} \quad (3.15)$$

Afin de minimiser J , on cherche simplement à annuler ses dérivées par rapport à u_k :

$$\begin{aligned} \frac{\partial J}{\partial U_k} = & \sum_{j \in I} U_j \int_{\mathcal{S}_T} \nabla \phi_j(\mathcal{X}) \cdot \nabla \phi_k(\mathcal{X}) \, ds + \sum_{j \in J} f(X_j) \int_{\mathcal{S}_T} \nabla \phi_k(\mathcal{X}) \cdot \nabla \phi_j(\mathcal{X}) \, ds \\ = & 0, \forall k \in I \end{aligned} \quad (3.16)$$

Il y a autant d'équations que de nœuds dans I . Pour résoudre les équations (3.16) avec des éléments finis linéaires, on peut calculer la matrice $A_{ij}^{\mathcal{T}_k}$ du triangle \mathcal{T}_k comme :

$$A_{ij}^{\mathcal{T}_k} = \int_{\mathcal{T}_k} \nabla \phi_i(\mathcal{X}) \cdot \nabla \phi_j(\mathcal{X}) \, ds = \int_0^1 \int_0^{1-\xi} \nabla^{\xi, \eta} \phi_i M_\xi^{-1} \nabla^{\xi, \eta} \phi_j \sqrt{\det M_\xi} \, d\xi d\eta \quad (3.17)$$

où $M_\xi = (\mathcal{X}_\xi)^t \mathcal{X}_\xi$ est le tenseur métrique de la paramétrisation $\mathcal{X}(\xi)$. Contrairement au cas continu, il a été montré que les cartes harmoniques discrètes ne sont pas toujours uniques.

3.2.3.2 Unicité des cartes harmoniques avec l'hypothèse de Floater

Pour assurer l'unicité des cartes harmoniques, Floater [Floater 1997] a utilisé la notion d'application barycentrique. Il a montré qu'il suffisait de vérifier les deux conditions suivantes :

1. l'image du bord de la surface \mathcal{S} par \mathcal{U} dans le domaine paramétrique est un polygone convexe ;
2. chaque nœud interne est une combinaison convexe de ses voisins.

Cette méthode est facilement codée en prenant pour chaque triangle \mathcal{T}_k :

$$A_{ij}^{\mathcal{T}_k} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \quad (3.18)$$

On retient que cette paramétrisation est beaucoup plus chaotique qu'avec une transformation harmonique classique.

3.2.3.3 Unicité des cartes harmoniques avec l'hypothèse de Remacle

Dans l'article [Remacle 2010], les auteurs proposent un algorithme appelé *cavity check* qui :

1. calcule la transformation harmonique en utilisant les éléments finis ;
2. vérifie pour chaque sommet à l'intérieur du domaine paramétrique que chacun des triangles voisins est orienté correctement ;
3. déplace le sommet incriminé des éléments retournés pour supprimer les chevauchements.

Cet algorithme est disponible dans le logiciel Gmsh[®] [Geuzaine 2009].

3.3 Calcul des sensibilités en utilisant des transformations harmoniques

3.3.1 Description théorique

Nous nous donnons un exemple pour expliquer notre démarche. Prenons un cylindre Ω_0 dont l'une des deux extrémités est fermée. Soit α_i un paramètre (par exemple, une dilatation). Soit Ω_{α_i} le cylindre obtenu après la modification du paramètre α_i . Soient $\tilde{\Omega}_0$ et $\tilde{\Omega}_{\alpha_i}$ les maillages associés (cf. figure 3.8).

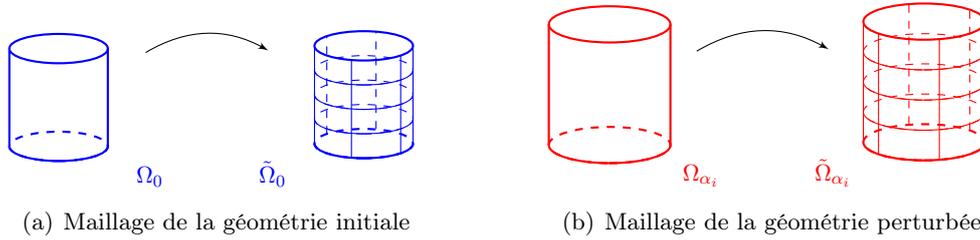


FIGURE 3.8 – Maillage des formes initiale et perturbée

On déplie chacun des deux maillages sur un disque unité en utilisant une transformation harmonique. Soient $\tilde{\mathcal{D}}_0$ et $\tilde{\mathcal{D}}_{\alpha_i}$ les deux disques associés à $\tilde{\Omega}_0$ et $\tilde{\Omega}_{\alpha_i}$. Soient \mathcal{U}_0 et \mathcal{U}_{α_i} les fonctions de passage (voir la figure 3.9) et \mathcal{U}_0^{-1} et $\mathcal{U}_{\alpha_i}^{-1}$ les fonctions de passage inverses.

Soient :

$$\begin{aligned} \mathcal{U}_0 : \tilde{\Omega}_0 \subset \mathbb{R}^3 &\longrightarrow \tilde{\mathcal{D}}_0(0,1) \\ (x,y,z) &\longmapsto \begin{cases} u_0(x,y,z) \\ v_0(x,y,z) \end{cases} \end{aligned} \quad (3.19)$$

$$\begin{aligned} \mathcal{U}_{\alpha_i} : \tilde{\Omega}_{\alpha_i} \subset \mathbb{R}^3 &\longrightarrow \tilde{\mathcal{D}}_{\alpha_i}(0,1) \\ (x,y,z) &\longmapsto \begin{cases} u_{\alpha_i}(x,y,z) \\ v_{\alpha_i}(x,y,z) \end{cases} \end{aligned} \quad (3.20)$$

Il suffit ensuite d'appliquer $\mathcal{U}_{\alpha_i}^{-1}$ sur $\tilde{\mathcal{D}}_0$. Ce nouveau maillage noté $\tilde{\Omega}_{0,\alpha_i}$ est la projection de $\tilde{\Omega}_0$ sur $\tilde{\Omega}_{\alpha_i}$, précisément ce que l'on recherchait. Pour cela, on interpole les nœuds du disque $\tilde{\mathcal{D}}_0$

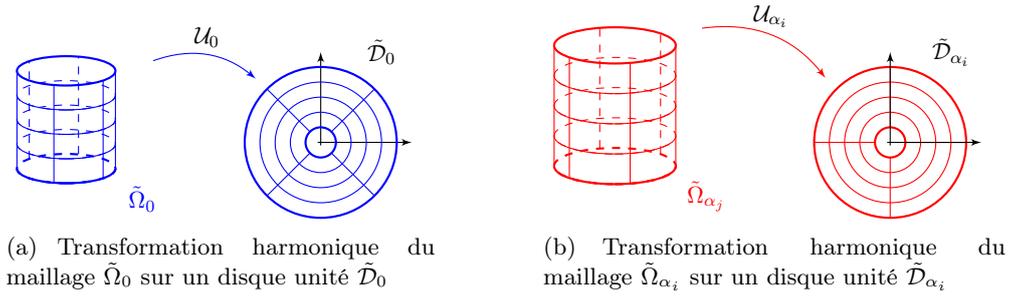


FIGURE 3.9 – Transformations harmoniques des maillages initial et perturbé

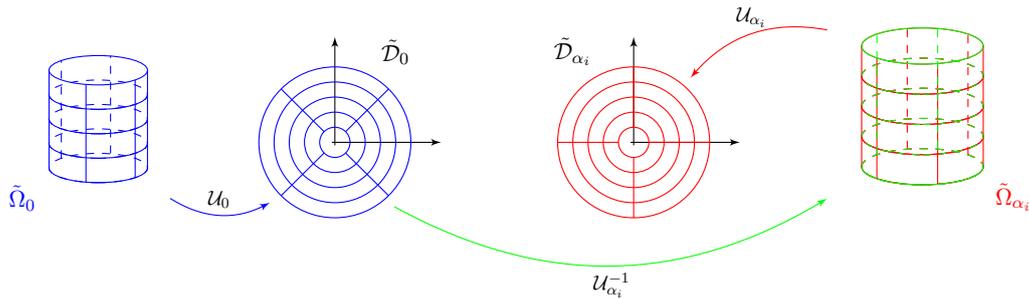


FIGURE 3.10 – Morphing du maillage de la forme initiale sur le maillage de la forme perturbée

avec ceux du disque $\tilde{\mathcal{D}}_{\alpha_i}$. Cette étape revient à « plaquer » le maillage de la forme initiale sur celui de la forme déformée (voir figure 3.10).

In fine, on connaît donc à ce stade les coordonnées des nœuds du maillage de la forme initiale $\tilde{\Omega}_0$ et leurs nouvelles coordonnées après avoir été morphées sur le maillage de la forme perturbée. On peut donc dorénavant évaluer l'influence de la perturbation par différences finies en utilisant l'équation (3.2) avec $\tilde{\Omega}(\alpha_i + \delta\alpha_i) = \mathcal{U}_{\alpha_i}^{-1}(\tilde{\mathcal{D}}_0(0,1))$. On obtient donc la sensibilité de la forme par rapport à chaque paramètre CAO par la formule (3.21) :

$$\partial_{\alpha_i} \Omega \simeq \frac{\mathcal{U}_{\alpha_i}^{-1}(\tilde{\mathcal{D}}_0(0,1)) - \tilde{\Omega}_0}{\delta\alpha_i} \quad i = 1 \dots p \quad (3.21)$$

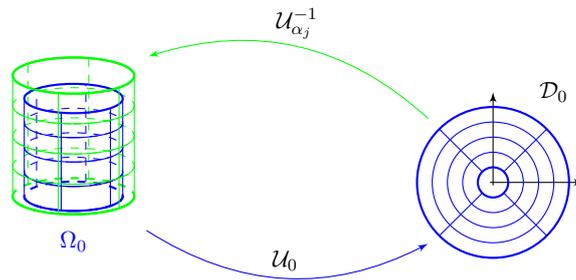


FIGURE 3.11 – Construction de $\tilde{\Omega}(\alpha_i + \delta\alpha_i)$

3.3.2 Procédure de calcul de la sensibilité

Nous suggérons la procédure suivante qui repose sur une transformation harmonique pour calculer la sensibilité d'un maillage par rapport à un paramètre.

Algorithme de calcul de la sensibilité par transformation harmonique – Soit Ω_0 une surface de \mathbb{R}^3 homéomorphe à un disque unité et α_i un paramètre de cette surface. Soit Ω_{α_i} une forme obtenue après la modification du paramètre α_i . Soient $\tilde{\Omega}_0$ et $\tilde{\Omega}_{\alpha_i}$ les maillages triangulaires associés. Soient $\tilde{\mathcal{D}}_0$ et $\tilde{\mathcal{D}}_{\alpha_i}$ les domaines paramétriques associés à $\tilde{\Omega}_0$ et $\tilde{\Omega}_{\alpha_i}$. L'algorithme de calcul de la sensibilité de $\tilde{\Omega}_0$ par rapport au paramètre α_i se décline en plusieurs étapes :

1. pour chacun des deux maillages $\tilde{\Omega}_0$ et $\tilde{\Omega}_{\alpha_i}$:
 - (a) lecture du maillage (nœuds et éléments) ;
 - (b) détection des arêtes du bord ;
 - (c) application des conditions limites de Dirichlet :
 - i. calcul de la longueur de chaque arête de la frontière ;
 - ii. calcul de la longueur L totale de la frontière ;
 - iii. application de l'équation (3.12).
 - (d) construction de la matrice des $A_{ij}^{\mathcal{T}_k}$:
 - soit en utilisant une transformation harmonique classique avec la formule (3.17) ;
 - soit en utilisant l'hypothèse de Floater avec la formule (3.18).
 - (e) résolution du problème mécanique (3.11).
2. interpolation des nœuds du disque $\tilde{\mathcal{D}}_0$ avec ceux de $\tilde{\mathcal{D}}_{\alpha_i}$.

Nous avons codé cet algorithme dans Scilab[®] 5.4.1 afin de l'appliquer sur différents cas d'étude.

3.3.3 Application sur un cas académique

La gestion des coins est essentielle pour le bon déroulement du calcul des sensibilités. C'est une des raisons pour laquelle on a proposé une nouvelle méthode. Nous prenons donc un cas académique : on translate un cube dans l'espace. Les résultats de la figure 3.12 montrent que notre méthode gère correctement les coins : on retrouve bien l'isoconnectivité des maillages. Les images ont été faites avec Scilab[®] 5.4.1.

3.4 Mise en place de la démarche

Dans cette partie, nous décrivons un cas d'étude sur lequel nous allons appliquer la démarche présentée ci-dessus.

3.4.1 Présentation du cas d'étude

On se donne une géométrie qui a la forme d'un coude à 90 degrés de section circulaire. L'entrée et la sortie restent ouvertes. Le coude est donc de genre 0 avec 2 frontières. Plusieurs sections sont réparties le long de la fibre neutre du coude (cf. figure 3.13(a)). On impose que la surface passe par ces sections.

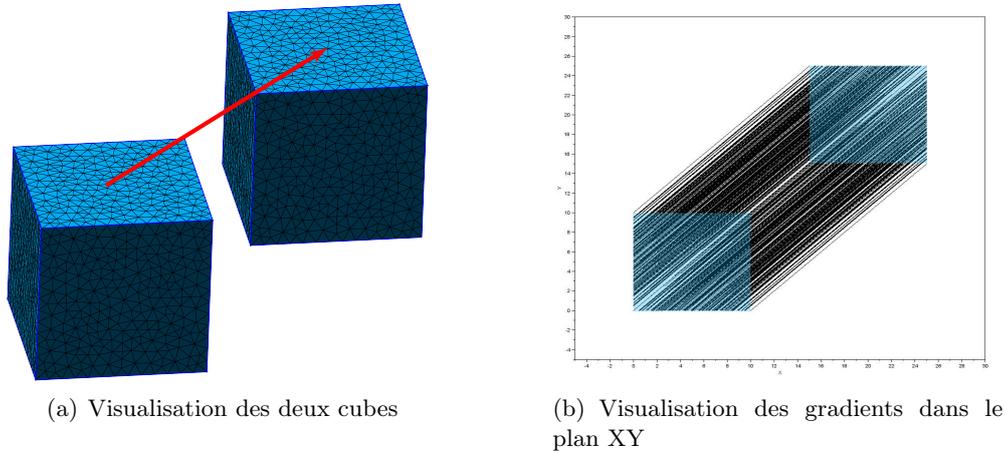


FIGURE 3.12 – Translation d'un cube et gradients associés

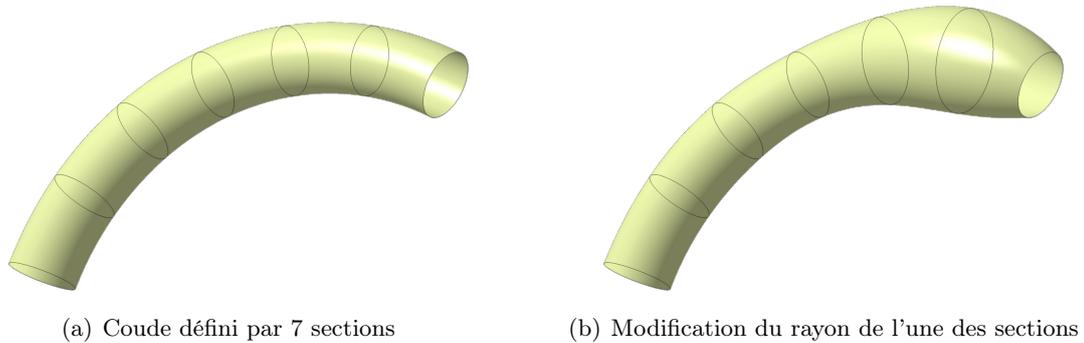


FIGURE 3.13 – Coude avec un paramètre

3.4.2 Choix du domaine paramétrique

Comme nous l'avons vu dans la partie 3.2.1 (p. 90), il est nécessaire que la surface \mathcal{S} de \mathbb{R}^3 ait la même topologie que le domaine paramétrique \mathcal{S}' de \mathbb{R}^2 afin que les deux surfaces soient homéomorphes. Notre cas d'étude est de genre 1 avec 2 frontières. On peut donc utiliser comme domaine paramétrique un disque contenant un trou (cf. figure 3.14).

On perturbe faiblement le coude initial afin d'obtenir une forme perturbée. On maille ensuite nos deux surfaces indépendamment l'une de l'autre avec des éléments triangulaires. Afin de mieux visualiser la perturbation, les maillages des deux coudes ont été superposés sur la figure 3.15.

L'objectif est de calculer le gradient associé à cette perturbation. Nous utilisons des transformations harmoniques basées sur l'algorithme *cavity check*, dont le code est disponible dans le logiciel Gmsh[®]. On impose des conditions de Dirichlet sur la frontière d'entrée et des conditions de Neumann sur celle de sortie (d'après l'équation (3.11)). Toutefois, l'évaluation du gradient se passe mal à cause du ratio entre la longueur de la frontière ∂S_1 et la longueur du coude. Le coude a une section d'entrée de 20 mm de rayon et une fibre neutre de 200 mm de rayon, soit un ratio d'environ $5,5\pi$. Or, d'après [Remacle 2010], les coordonnées paramétriques calculées deviennent indifférenciables à cause de la précision machine au delà d'un certain ratio (aux alentours de 6π

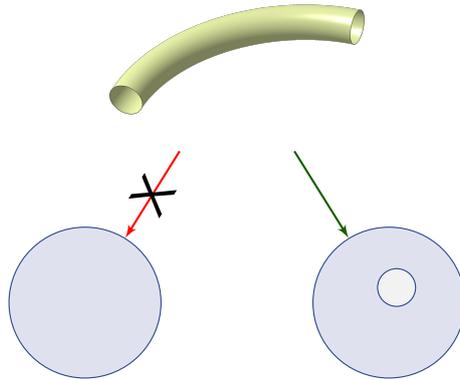


FIGURE 3.14 – Choix du domaine paramétrique

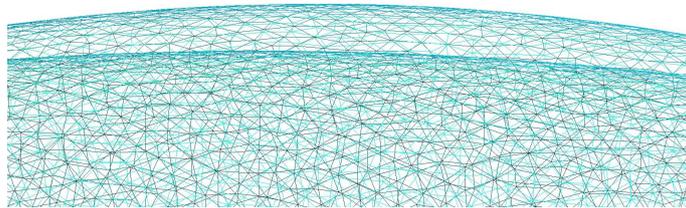


FIGURE 3.15 – Comparaison des maillages du coude initial et du coude perturbé

en double précision).

Pour contourner ce problème, nous proposons de découper les géométries afin de n'avoir qu'une seule frontière. Comme le coude est symétrique, chaque pièce est coupée en deux suivant le plan de symétrie. Chaque demi-coude est donc homéomorphe à un disque.

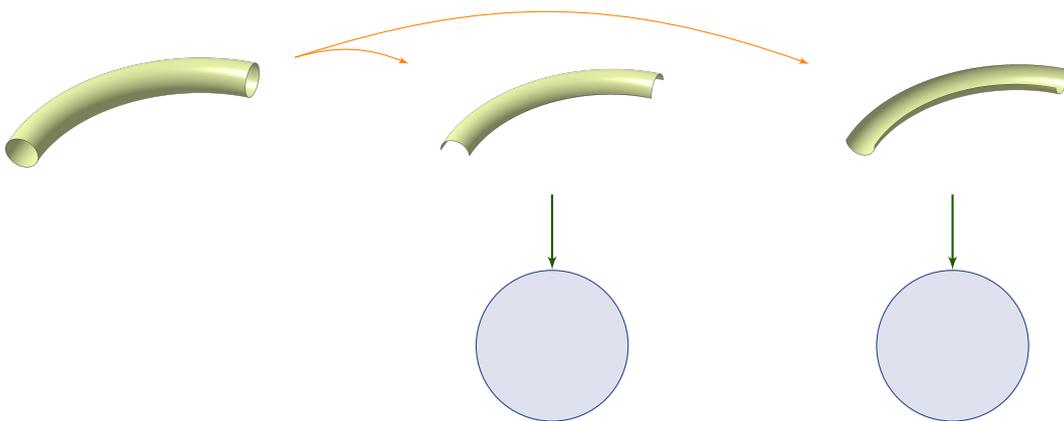


FIGURE 3.16 – Découpage de la géométrie en deux parties homéomorphes à un disque

L'évaluation du gradient se fait sur chaque demi-partie. On exporte les coordonnées paramétriques (u_i, v_i) de chaque nœud (x_i, y_i, z_i) du maillage initial.

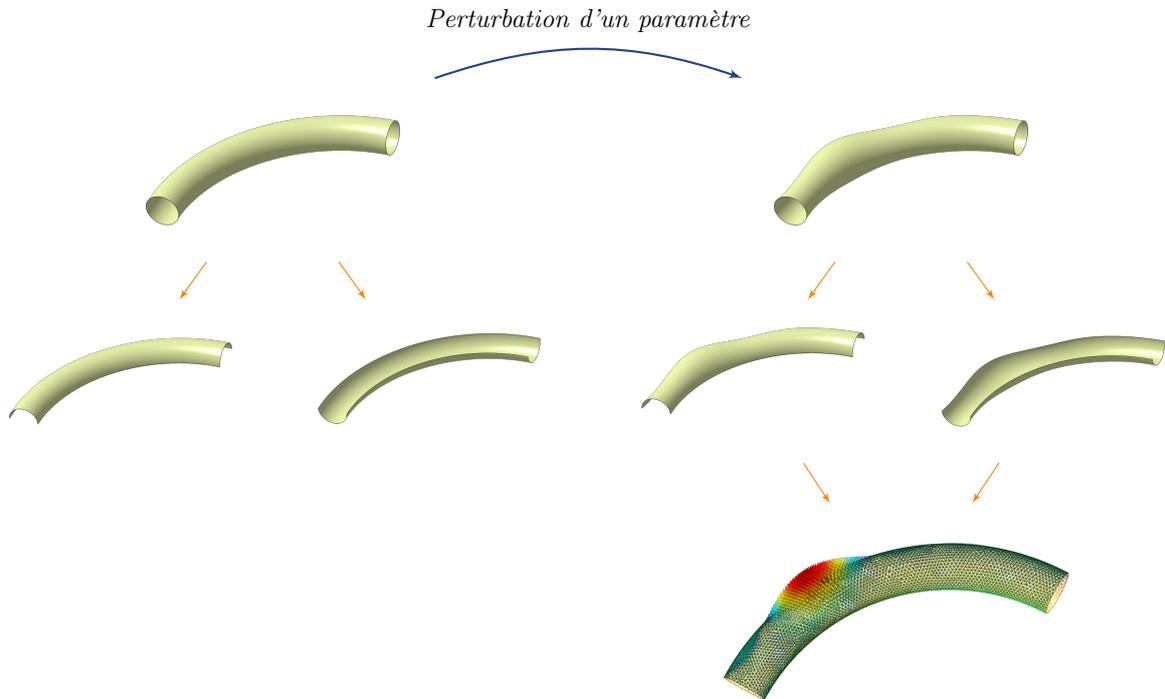


FIGURE 3.17 – Description schématique de l'évaluation des sensibilités

3.4.3 Point invariant en 2D et sens de parcours de la frontière

Afin de pouvoir interpoler nos deux disques, il est nécessaire de définir un point invariant sur le domaine paramétrique et de parcourir la frontière ∂S_1 dans le même sens. Sans cela, le choix du sommet initial et de l'orientation des disques est laissé au hasard, ce qui entraîne des rotations des disques paramétriques. La figure 3.18 illustre la rotation entre les domaines paramétriques de la forme initiale et de la forme perturbée. Elle a été obtenue en utilisant la fonction d'interpolation cubique basée sur la méthode de Shepard *cshep2d* de Scilab[®] 5.4.1. Chaque trait noir correspond au déplacement d'un nœud entre la forme nominale et perturbée.

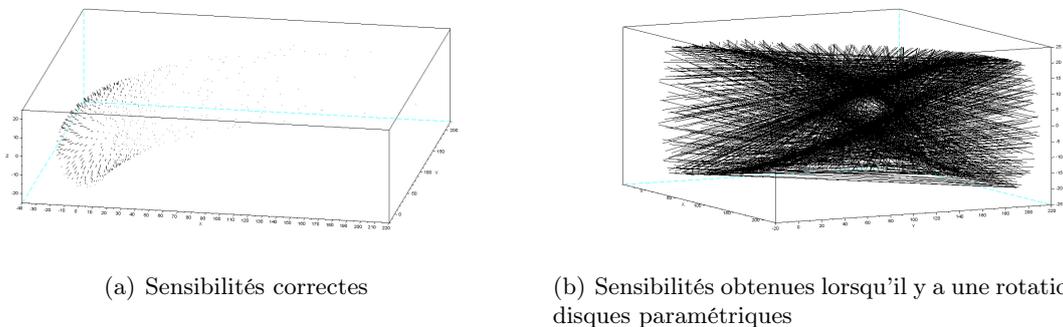


FIGURE 3.18 – Rotation des disques paramétriques

Le logiciel Gmsh[®] prend arbitrairement comme sommet initial le premier point rencontré

(dans le fichier de maillage) situé sur la frontière ∂S_1 . Les maillages des formes nominale et perturbée sont non structurés. Afin d'imposer le point invariant en 2D, nous définissons dans la CAO un point de référence situé sur la frontière ∂S_1 ainsi que le sens de parcours de cette frontière. Nous récupérons ces informations dans le logiciel de maillage pour identifier quel nœud du maillage correspond au point de référence. L'identifiant de ce nœud est ensuite fixé à 1. Pour imposer le sens de parcours de la frontière, on détermine quel nœud est rencontré le long de ∂S_1 en suivant le sens imposé dans la CAO. On fixe l'identifiant de ce nœud à 2. Ainsi, la géométrie initiale et les géométries perturbées ont toutes un nœud en commun et un sens de parcours de la frontière du maillage imposé.

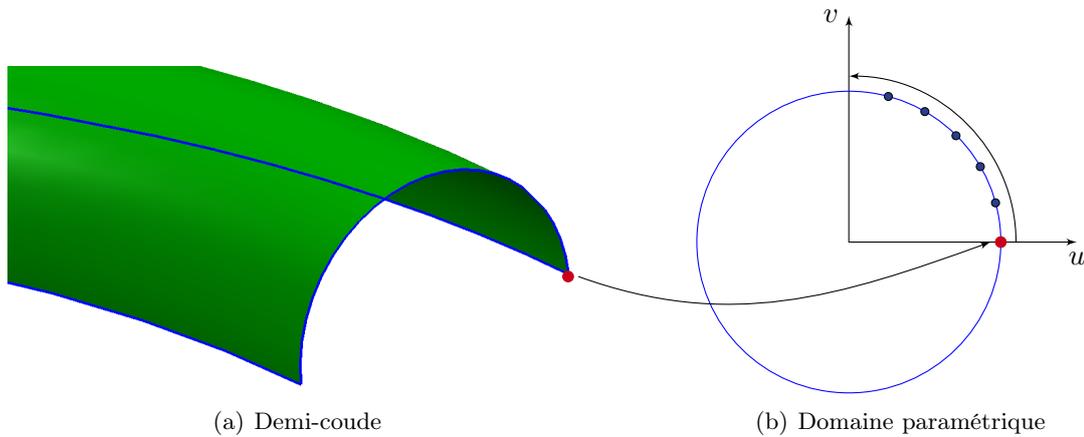


FIGURE 3.19 – Paramétrisation du demi-coude sur un disque unité : le point invariant 2D est représenté par le point rouge. La frontière ∂S_1 du demi-coude est en bleu.

Ce qui précède nous permet de déterminer les sensibilités du maillage par rapport aux paramètres CAO. Le gradient est déterminé pour chaque partie du coude Ω_a et Ω_b . Les gradients trouvés sont cohérents avec la perturbation imposée dans le logiciel CAO. Une visualisation de ces gradients est donnée par la figure 3.20. Nous nous sommes appuyés sur les outils de post-traitement de Gmsh[®] 2.8.3 pour implémenter la visualisation de ces sensibilités au sein de ce logiciel.

3.4.4 Calcul du gradient au niveau des nœuds ou des éléments

Pour n'avoir qu'une seule frontière, on découpe notre géométrie en parties ($\Omega = \Omega_a \cup \Omega_b \cup \Omega_c \cup \dots$). Chaque partie représente donc une partition des éléments. Au niveau de la couture entre les parties, les gradients du maillage par rapport aux paramètres CAO sur les nœuds ne sont pas identiques, ce qui pourrait induire des erreurs lors du calcul de sensibilité. Ce comportement est normal : il est lié à l'augmentation ou à la réduction du périmètre d'un patch après application de la perturbation. Il est possible de :

- calculer le gradient de la prestation par rapport aux paramètres CAO sur chaque partie $\{\Omega_a, \Omega_b, \Omega_c \dots\}$ et de sommer les contributions de chaque partie,
- calculer le gradient au niveau des centroïdes des éléments : il faut donc calculer la contribution de chaque élément pour la sensibilité de la performance par rapport au maillage

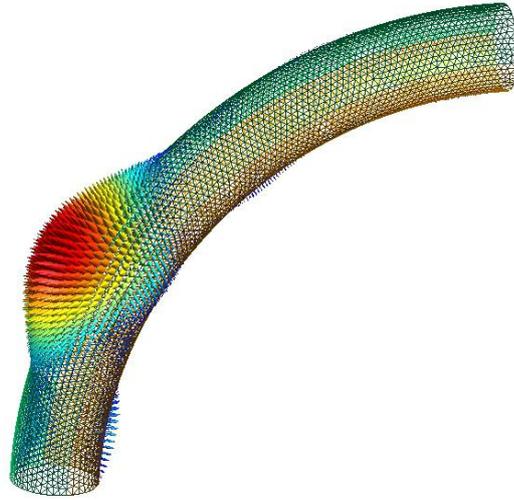


FIGURE 3.20 – Visualisation des gradients après modification des outils de post-traitement de Gmsh[®] 2.8.3

et estimer la sensibilité du maillage par rapport aux paramètres CAO au niveau des centroïdes.

3.4.4.1 Calcul des sensibilités au niveau des nœuds du maillage

Au début de cette partie, on a proposé de calculer $\partial_{\alpha_i} J$ par l'équation (3.1). En passant à sa version discrète, on a :

$$d_{\alpha_i} J = \partial_{\tilde{\Omega}} J \times \partial_{\alpha_i} \tilde{\Omega} \quad i = 1 \dots p \quad (3.22)$$

On décompose ce calcul sur chaque partie $\{\Omega_a, \Omega_b, \Omega_c \dots\}$:

$$\begin{aligned} \frac{dJ}{d\alpha_i} &= \frac{\partial J}{\partial \tilde{\Omega}} \cdot \frac{\partial \tilde{\Omega}}{\partial \alpha_i} \quad i = 1 \dots p \\ &= \sum_{k=a,b,c,\dots} \frac{\partial J}{\partial \tilde{\Omega}_k} \cdot \frac{\partial \tilde{\Omega}_k}{\partial \alpha_i} \quad i = 1 \dots p \\ &= \sum_{k=a,b,c,\dots} \left(\sum_{x_j \in \tilde{\Omega}_k} \frac{\partial J}{\partial x_j} \cdot \frac{\partial x_j}{\partial \alpha_i} \right) \quad i = 1 \dots p \end{aligned} \quad (3.23)$$

en notant x_j la position d'un nœud du maillage.

3.4.4.2 Calcul des sensibilités au niveau des centroïdes des éléments

Certains codes renvoient le gradient par rapport aux centroïdes des éléments, notamment en mécanique des fluides. Les formules vont alors s'écrire (en notant x_e le centroïde de l'élément e) :

$$\begin{aligned}
 \frac{dJ}{d\alpha_i} &= \frac{\partial J}{\partial \tilde{\Omega}} \cdot \frac{\partial \tilde{\Omega}}{\partial \alpha_i} \quad i = 1 \dots p \\
 &= \sum_{k=a,b,c,\dots} \frac{\partial J}{\partial \tilde{\Omega}_k} \cdot \frac{\partial \tilde{\Omega}_k}{\partial \alpha_i} \quad i = 1 \dots p \\
 &= \sum_{k=a,b,c,\dots} \left(\sum_{e \in \tilde{\Omega}_k} \frac{\partial J}{\partial x_e} \cdot \frac{\partial x_e}{\partial \alpha_i} \right) \quad i = 1 \dots p
 \end{aligned} \tag{3.24}$$

dans lesquelles on pose : $\frac{\partial x_e}{\partial \alpha_i} = \left(\sum_{j \in \mathcal{T}_e} \frac{\partial x_j}{\partial \alpha_i} \right) \times \frac{1}{\#_e}$ avec $\#_e$ le nombre de nœuds appartenant à l'élément e .

On pourrait également faire le calcul aux centroïdes quand bien même $\partial_{\tilde{\Omega}_k} J$ serait donné aux nœuds :

1. calcul d'une densité de contribution en chaque nœud x_i :

$$c_i = \frac{\partial J}{\partial x_i} \times \frac{1}{\mathcal{A}_i}$$

avec $\mathcal{A}_i = \sum_{e \ni i} \frac{\mathcal{A}_e}{\#_e}$ et \mathcal{A}_e l'aire de l'élément e .

2. calcul d'une densité de contribution en chaque centroïde :

$$c_e = \frac{1}{\#_e} \sum_{i \in e} c_i$$

3. calcul de la contribution de chaque élément :

$$\frac{\partial J}{\partial x_e} = c_e \cdot \mathcal{A}_e$$

3.4.5 Valeurs des perturbations

Les systèmes CAO sont capables de travailler avec des précisions plus ou moins grandes en fonction de la taille des modèles (voir le tableau 3.1).

Les industriels travaillent généralement en échelle standard. Les valeurs des perturbations appliquées à nos paramètres seront donc détectables par les logiciels CAO. Il faudra alors s'assurer que la perturbation appliquée à un paramètre engendre bien des modifications géométriques identifiables par le maillage et qui permettent de calculer convenablement la sensibilité du maillage.

Échelle	Majorité des éléments supérieurs à	Majorité des éléments inférieurs à	Précision
	[mm]	[mm]	[mm]
Petite	10^{-3}	10^2	10^{-5}
Standard	10^{-1}	10^5	10^{-3}
Grande	10	10^6	10^{-1}

Tableau 3.1 – Précisions des systèmes CAO en fonction de la taille des modèles

3.5 Application de la méthode

3.5.1 Boucle d'optimisation

Ce chapitre a permis de lever le verrou lié au calcul des sensibilités du maillage par rapport aux paramètres CAO. Il faut maintenant assembler l'ensemble des briques dans une boucle d'optimisation directe. Nous utilisons un algorithme du gradient sans contrainte proche de celui présenté dans la partie 1.3.3.1 (p. 26). Le pilotage de la boucle et des logiciels qui interviennent a été réalisé avec Scilab® 5.4.1. Ci-dessous nous détaillons les différentes étapes de la figure 3.21.

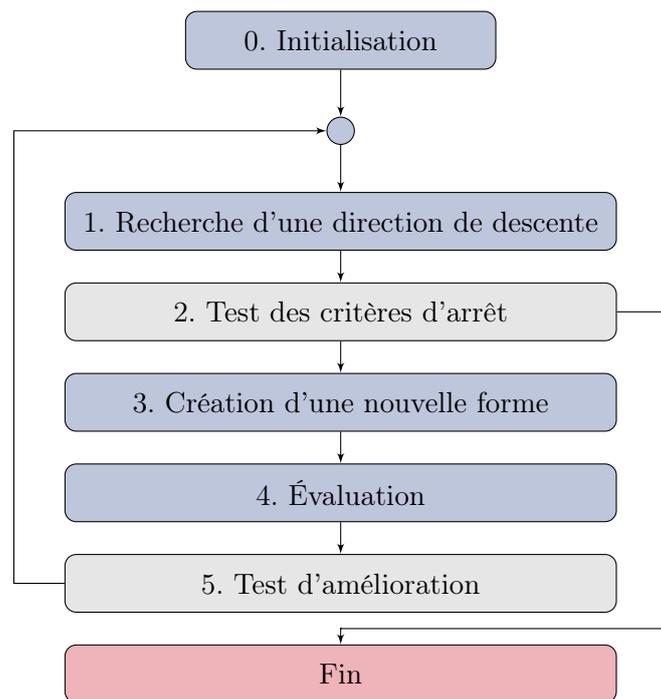


FIGURE 3.21 – Schéma de la boucle d'optimisation directe

0. Initialisation

La phase d'initialisation comprend les étapes suivantes :

- lecture des valeurs initiales des paramètres et de leurs bornes ;

- création de la table des géométries comprenant la géométrie initiale Ω_0 et ses perturbations par chaque paramètre ;
- génération des géométries ;
- maillage de Ω_0 et mise en données ;
- maillage des formes perturbées de Ω_0 ;
- évaluation de la solution initiale :
 - évaluation de la performance J (solveur primaire) ;
 - évaluation du gradient de J par rapport au maillage, *i.e.* $\partial_{\tilde{\Omega}_0} J$ (solveur adjoint) ;
 - évaluation de la sensibilité du maillage par rapport aux paramètres CAO, *i.e.* $\partial_{\alpha_i} \tilde{\Omega}_0$.

1. Recherche d'une direction de descente

La fonction J décroît, dans le voisinage du paramètre α_i , le plus fortement dans la direction opposée à celle du gradient de J en α_i , à savoir dans la direction $-\nabla J(\alpha_i)$. Les sensibilités sont calculées pour chaque paramètre par l'équation (3.22) : $\partial_{\alpha_i} J = \partial_{\tilde{\Omega}} J \times \partial_{\alpha_i} \tilde{\Omega}$.

Soit V la direction de descente :

$$V = \begin{cases} +\nabla J & \text{si l'on cherche à maximiser la fonction objectif} \\ -\nabla J & \text{si l'on cherche à minimiser la fonction objectif} \end{cases}$$

en prenant la notation conventionnelle $\nabla J = \begin{pmatrix} \partial_{\alpha_1} J \\ \partial_{\alpha_2} J \\ \dots \\ \partial_{\alpha_p} J \end{pmatrix}$.

2. Test des critères d'arrêt

Nous avons utilisé trois critères d'arrêt pour la boucle :

- le premier porte sur le pas : $\beta < 10^{-4}$;
- le deuxième concerne l'évolution de J à l'itération k : $\left| \frac{J^{\text{test}} - J^k}{J^0} \right| < 10^{-5}$;
- le troisième relève de la vitesse effective : $\left\| \frac{\alpha^{\text{test}} - \alpha^k}{\beta} \right\| < 10^{-5}$.

3. Création d'une nouvelle forme

On utilise ensuite la direction de descente V pour mettre à jour les valeurs des paramètres α_i :

$$\alpha^{\text{test}} = \alpha^k + \beta V \tag{3.25}$$

avec α^{test} les nouvelles valeurs des paramètres CAO. On s'assure que ces nouvelles valeurs sont comprises dans les plages de variation autorisées.

4. Évaluation de la forme

Cette étape est quasiment analogue à celle de l'initialisation :

- génération des géométries ;
- maillage de Ω_{test} et mise en données ;
- maillage des formes perturbées de Ω_{test} ;
- évaluation de la solution initiale.

5. Test d'amélioration

La performance à l'itération actuelle J^{test} est comparée avec la meilleure valeur obtenue J^k .
 Si $J^{\text{test}} < J^k$:

- $\beta = 1,1 \times \beta$;
- $\alpha^k = \alpha^{\text{test}}$;
- $J^k = J^{\text{test}}$;
- $\nabla J^k = \nabla J^{\text{test}}$.

Sinon, on divise le pas β par 2.

3.5.2 Cas d'étude

Nous reprenons la géométrie présentée dans la partie 3.4.1 (p. 97). Nous allons améliorer petit à petit le paramétrage de la géométrie. Nous utilisons les solveurs adjoints en mécanique des solides. Notre objectif est de valider la brique de calcul des sensibilités par rapport aux paramètres CAO.

La boucle a été développée sous Scilab[®] 5.4.1. La CAO est traitée avec le logiciel CATIA[®] V5. On y intègre notamment le point invariant du domaine paramétrique et le sens de parcours de la frontière. Les maillages sont réalisés avec ANSA (édité par BETA CAE Systems) ainsi que la mise en données. Enfin, les sensibilités mécaniques sont fournies par le logiciel Nastran[®] (en utilisant la carte *sol200*). L'épaisseur de la surface est fixée à 1 mm et la carte matériau est définie avec les valeurs suivantes :

- module de Young : $E = 7,4 \times 10^7 \frac{kg}{mm.s^2}$;
- coefficient de Poisson : $\nu = 0.33$;
- densité : $\rho = 2,7 \times 10^{-6} \frac{kg}{mm^3}$.

Dans l'état de l'art (voir la partie 1.3 (p. 16)), nous avons présenté deux problèmes de conception optimale : l'un en raideur statique et l'autre en vibratoire. Nous les reprenons ici.

3.5.2.1 Cahier des charges en NVH

On cherche à maximiser la pulsation ω_1 du premier mode propre du coude, ce qui équivaut à minimiser $J(\omega(\alpha)) = -\omega_1^2$. On a :

$$\left\{ \begin{array}{l} \min_{\alpha} J(\omega(\alpha)) \\ s.t. \underline{\alpha}_i \leq \alpha_i \leq \bar{\alpha}_i \quad \forall i = 1 \dots p \end{array} \right. \tag{3.26}$$

On note $\omega(\alpha) \geq 0$ la pulsation propre et $\phi(\alpha)$ le mode associé (c'est-à-dire le champ de déplacement correspondant sur Ω) qui sont solutions du problème suivant :

$$\left\{ \begin{array}{ll} -\text{div } \sigma(\phi) = \omega^2 \rho \phi & \text{sur } \Omega \\ \phi = 0 & \text{sur } \Gamma_D \\ \sigma(\phi)n = 0 & \text{sur } \Gamma_N \cup \Gamma \end{array} \right. \tag{3.27}$$

avec ρ la densité volumique du matériau et σ le tenseur des contraintes qui suit la loi de Hooke. (3.27) admet une infinité de solutions $(\omega_k, \phi_k)_{k \geq 1}$ que l'on classe dans l'ordre croissant des pulsations.

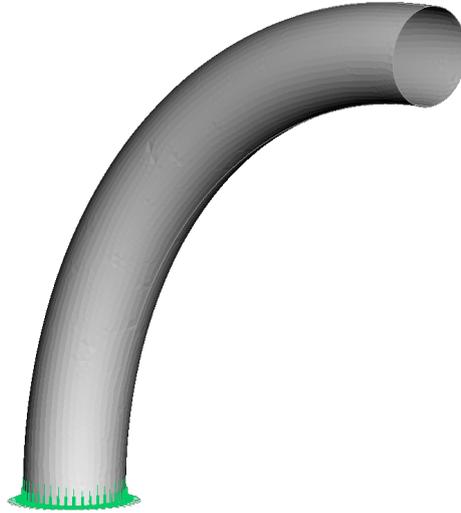


FIGURE 3.22 – Encastrement de la base du coude

1^{re} salve de résultats

Le coude comporte 7 sections circulaires. Chacune des sections est pilotée par deux paramètres qui permettent d'ajuster le rayon et de décentrer la section par rapport à la fibre neutre comme le montre la figure 3.23. Le décentrement de la section de base est bloqué. Cela fait donc 13 paramètres en tout. Nous avons défini une surface dans l'environnement CAO qui coupe la géométrie en deux parties à peu près équivalentes (cf. figure 3.24).

Pour une taille de maille de 2 mm, le maillage contient 11100 nœuds et 22100 triangles, contre 5000 nœuds et 9900 triangles pour une taille de 3 mm. L'évaluation d'une itération complète prend environ 21 minutes pour une taille de maille de 2 mm, contre 12 minutes pour une taille de 3 mm. On conserve le même mode (flexion) tout au long de l'optimisation. Le tableau 3.2 reprend l'ensemble des résultats :

1. la comparaison des deux premiers cas nous montre que l'on tombe dans deux minima locaux lorsque que l'on part de solutions initiales différentes ;
2. au regard du premier et du troisième cas, effectuer le calcul au niveau des nœuds ou des centroïdes n'a pas d'influence sur les résultats ;
3. le premier et le quatrième cas ne montrent pas d'incidence sur les résultats lorsque l'on diminue la taille de maille.

Pour les cas n° 1, 3 et 4, les bornes ont été atteintes par 9 des 13 paramètres et pour le cas n° 2 par 8 paramètres (voir la figure 3.25(b)). La forme optimisée (cf. figure 3.26) est atypique mais l'historique de la fonction objectif (voir la courbe 3.25(a)) montre que l'on a au final trois fois la première fréquence propre. On a d'ailleurs essayé d'optimiser nous-même le tube en définissant les valeurs des paramètres mais le résultat est beaucoup moins bon. On peut néanmoins se rassurer sur la forme en disant que l'algorithme a cherché à réduire la masse en porte-à-faux (d'où le bec)

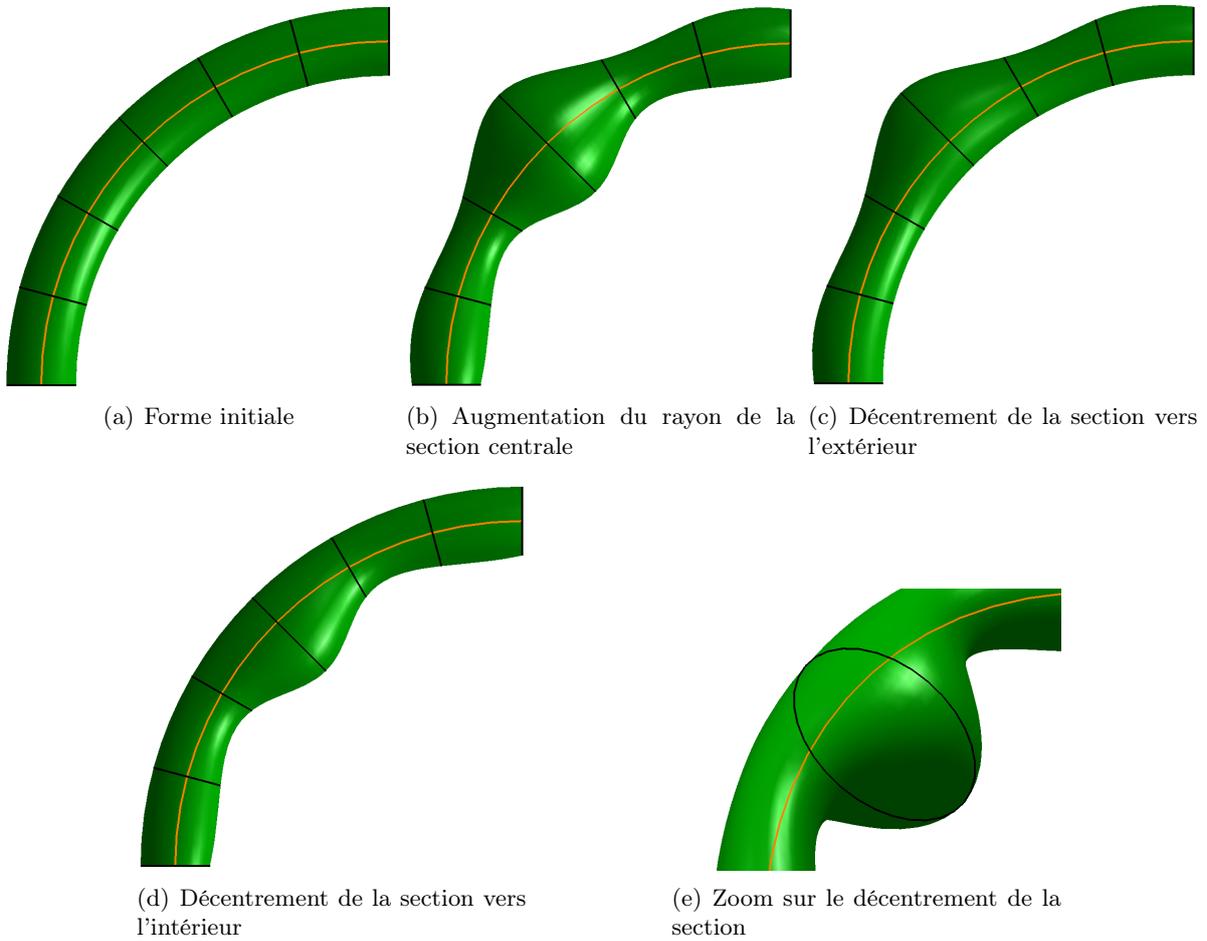


FIGURE 3.23 – Paramétrage du coude avec 7 sections : les sections sont en noir et la fibre neutre en orange.

Cas	Calcul des gradients aux	Valeurs initiales [mm]	Fréquence initiale [Hz]	Fréquence optimisée [Hz]	Nombre d'itérations	Taille de maille [mm]
1	nœuds	20	306,20	1007,60	36	2
2	nœuds	35	379,59	893,41	64	2
3	centroïdes	20	306,20	1001,20	40	2
4	nœuds	20	306,61	1008,64	53	3

Tableau 3.2 – Maximisation de la fréquence du premier mode propre : le modèle comporte 13 paramètres dont les plages de variation sont comprises entre 10 et 40 mm. Le décentrement des sections circulaires par rapport à la fibre neutre est autorisé.

et à élargir la base pour gagner en raideur. Le décrochement au centre semble apporter de la raideur en flexion.

Remarque : sur la figure 3.27 est tracée la sensibilité du maillage par rapport à un paramètre. De manière générale, on observe que certaines sensibilités peuvent être fortes au niveau de certains

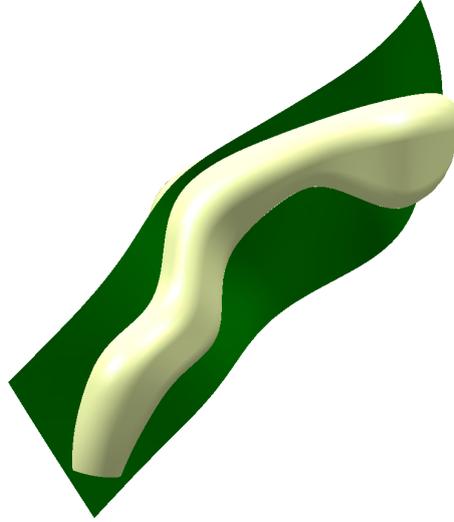
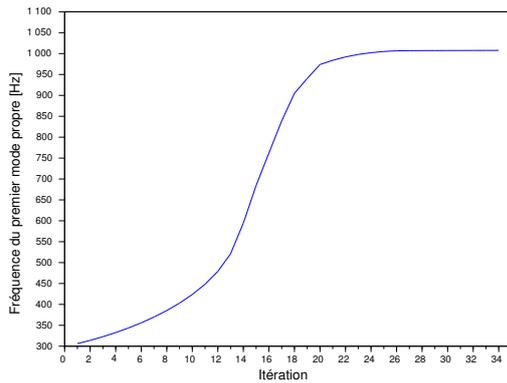
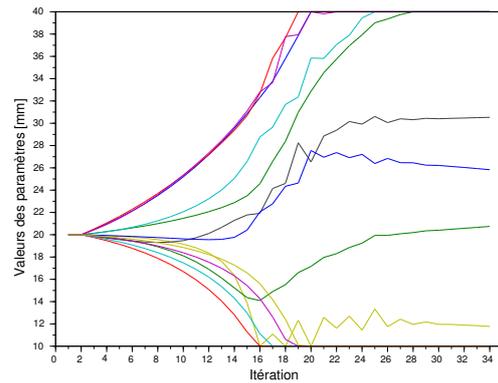


FIGURE 3.24 – Surface de découpe de la géométrie



(a) Évolution de la fréquence au fil des itérations



(b) Évolution de la valeur des paramètres au fil des itérations

FIGURE 3.25 – Courbes des résultats du cas n° 1

nœuds, plus particulièrement au niveau des angles des différentes parties (Ω_a , Ω_b ...). Cela est certainement dû à l'écrasement de certains éléments lors de la paramétrisation harmonique. Quand on raffine le maillage, ces valeurs ont tendance à disparaître.

2^e salve de résultats

Nous avons dans un second temps changé totalement le paramétrage de notre coude en prenant des sections rectangulaires. À nouveau, on observe (voir la figure 3.28) un resserrement des sections finales pour limiter la matière en porte-à-faux : les paramètres sont au minimum. À la base, au contraire, les sections sont au maximum. Entre les deux, elles varient continûment comme on l'aurait fait à la main.

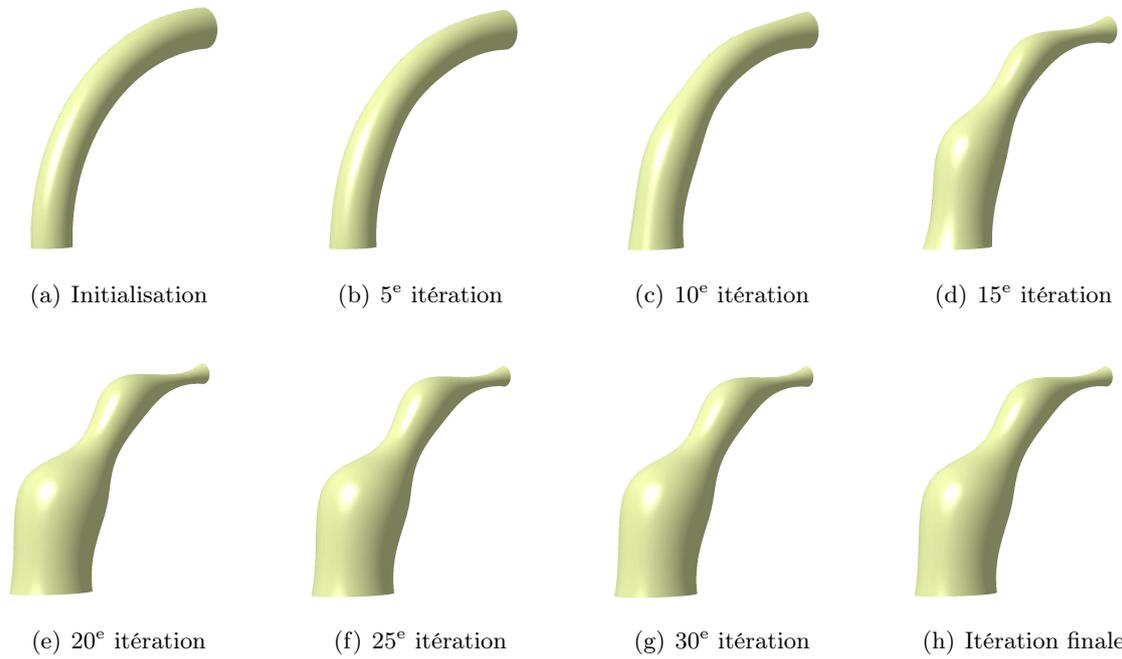


FIGURE 3.26 – Évolution de la forme du coude à sections circulaires au fil des itérations (cas n° 1)

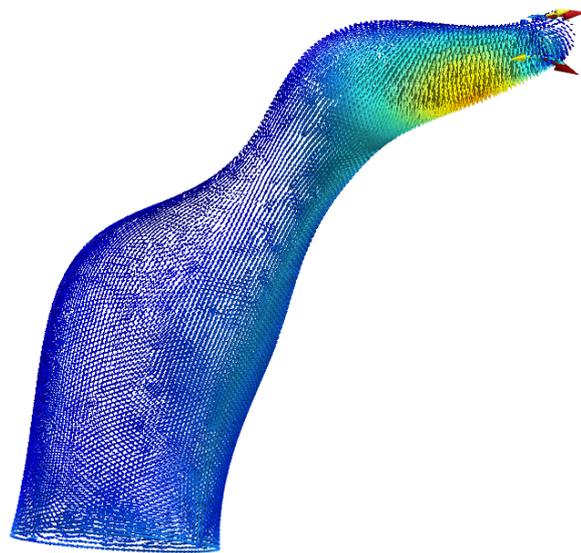


FIGURE 3.27 – Visualisation de la sensibilité du maillage par rapport à un paramètre CAO (itération finale du cas n° 1) : certaines valeurs de gradient sont fortes comparées aux valeurs voisines.

Cas	Calcul des gradients aux	Fréquence initiale [Hz]	Fréquence optimisée [Hz]	Nombre d'itérations	Taille de maille [mm]
5	nœuds	217,07	671,37	58	2
6	nœuds	217,59	673,87	52	3

Tableau 3.3 – Maximisation de la fréquence du premier mode propre : le modèle comporte 14 paramètres. Il n'y a pas de décentrement des sections.

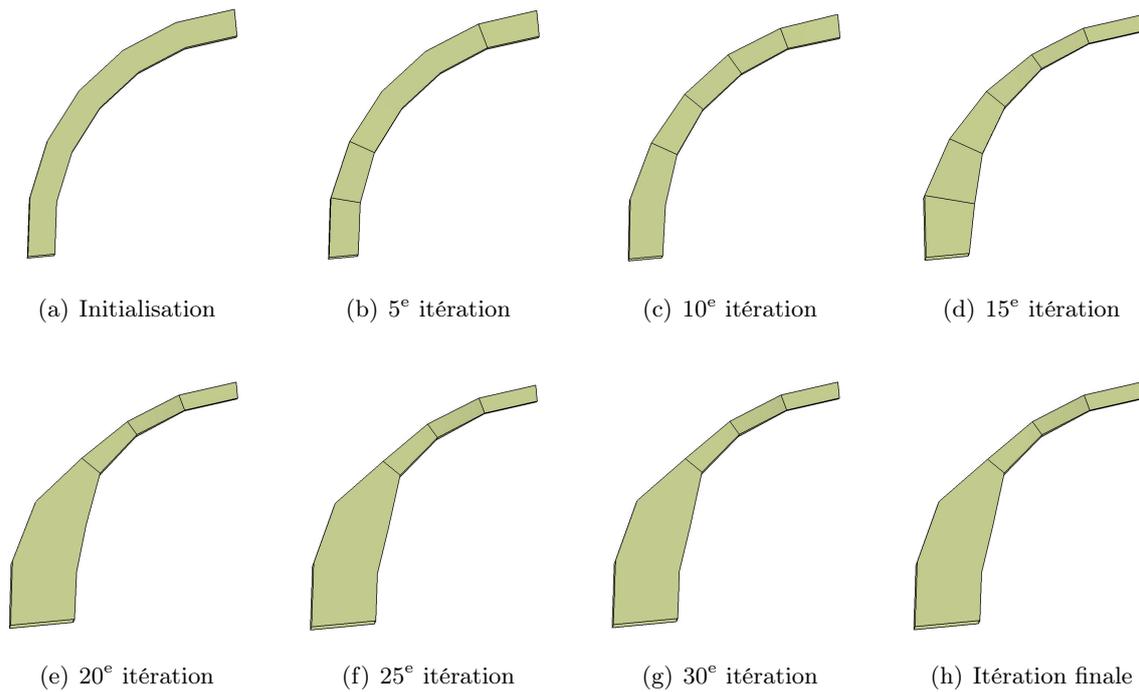


FIGURE 3.28 – Évolution de la forme du coude à sections rectangulaires au fil des itérations (cas n° 5)

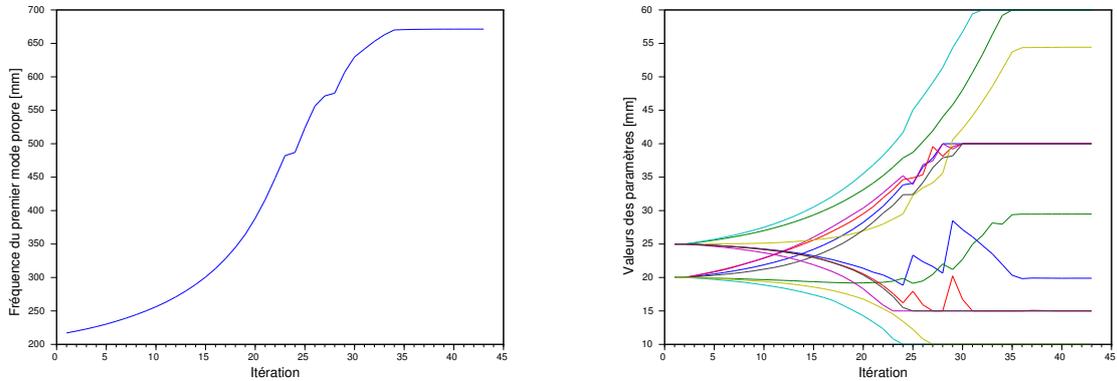
3.5.2.2 Cahier des charges en raideur

On charge le nœud maître du réseau de RBE2 qui permet de distribuer la force sur la section finale (voir la figure 3.30). La base du coude est fixée. Pour maximiser sa raideur, on va essayer de contrôler le déplacement u_m du nœud maître que l'on souhaite le plus faible possible. On pose $J(\alpha) = \|u_m(\alpha)\|$. On a :

$$\begin{cases} \min_{\alpha} J(\alpha) \\ \text{s.t. } \underline{\alpha}_i \leq \alpha_i \leq \bar{\alpha}_i \quad i = 1 \dots p \end{cases} \quad (3.28)$$

avec $u(\alpha)$ solution du problème d'élasticité linéaire suivant :

$$\begin{cases} -\operatorname{div} \sigma(u) = 0 & \text{dans } \Omega \\ u = 0 & \text{sur } \Gamma_D \\ \sigma(u) \cdot n = g & \text{sur } \Gamma_N \\ \sigma(u) \cdot n = 0 & \text{sur } \Gamma \end{cases} \quad (3.29)$$



(a) Évolution de la fréquence au fil des itérations (b) Évolution de la valeur des paramètres au fil des itérations

FIGURE 3.29 – Courbes des résultats du cas n° 5

avec g le vecteur des forces appliquées (dans notre cas, une force ponctuelle au nœud maître), n le vecteur unitaire normal à $\partial\Omega$ et σ le tenseur des contraintes qui suit la loi de Hooke.

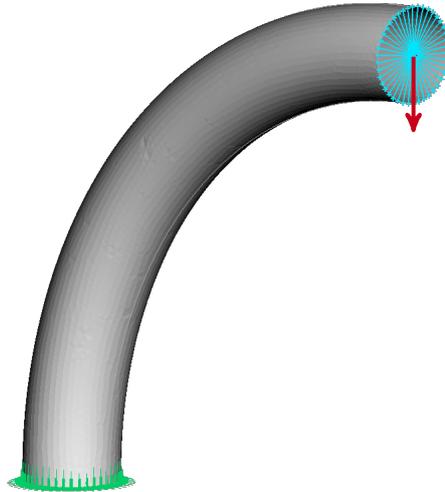
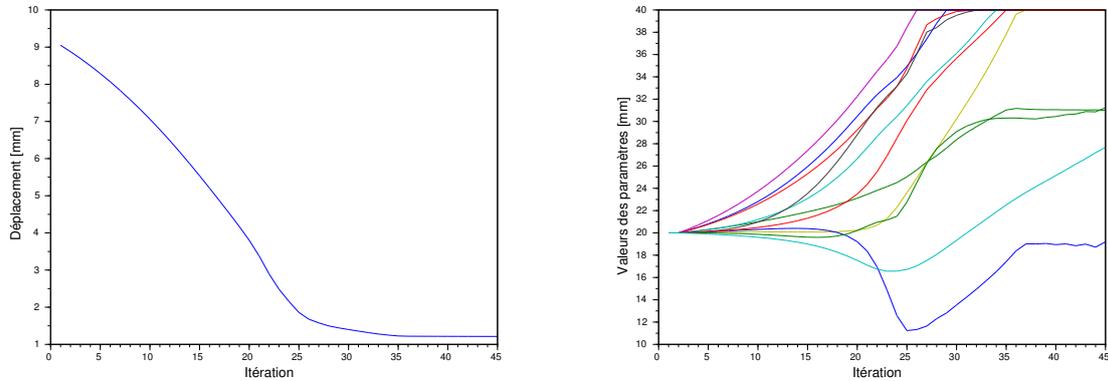


FIGURE 3.30 – Mise en données du problème en raideur statique : encastrement de la base et création du réseau d'éléments rigides.

1^{re} salve de résultats

On utilise le même paramétrage que pour les cas n° 1 à 4. On bloque les deux paramètres de la section finale où s'applique le réseau de RBE2. Il y a donc 11 paramètres. On constate un facteur 7 entre le déplacement initial et final du nœud maître (voir le tableau 3.4). 7 paramètres ont atteint les bornes. La forme optimisée est de nouveau atypique comme on peut le visualiser sur la figure 3.32.



(a) Évolution du déplacement au fil des itérations (b) Évolution de la valeur des paramètres au fil des itérations

FIGURE 3.31 – Courbes des résultats du cas n° 7

Cas	Calcul des gradients aux	Déplacement initial [mm]	Déplacement optimisé [mm]	Nombre d'itérations	Taille de maille [mm]
7	nœuds	9,05	1,21	45	2
8	nœuds	9,03	1,21	49	3

Tableau 3.4 – Minimisation du déplacement : le modèle comporte 11 paramètres dont les plages de variation sont comprises entre 10 et 40 mm. Le décentrement des sections circulaires par rapport à la fibre neutre est autorisé. Les paramètres sont initialisés à 20 mm.

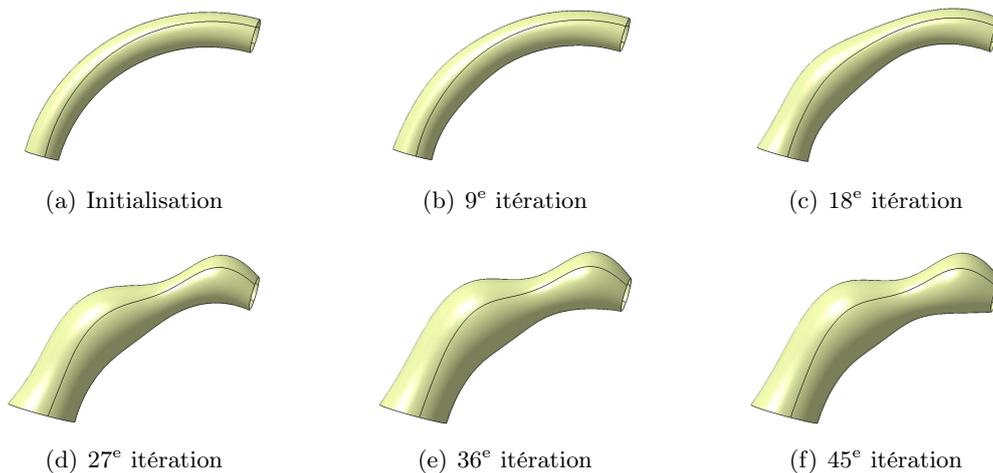


FIGURE 3.32 – Évolution de la forme du coude à sections circulaires au fil des itérations (cas n° 7)

2^e salve de résultats

On améliore la flexibilité du coude en ajoutant des paramètres qui permettent de passer de sections circulaires à des sections rectangulaires de manière « continue ». Qui plus est, pour le cas

Chapitre 3. Calcul de la sensibilité d'une prestation par rapport aux paramètres métier

n° 12, on ajoute des sections et on les autorise à glisser le long de la fibre neutre. Cette évolution du paramétrage est présentée sur les figures 3.33 et 3.34. Pour les cas n° 9 à 11, chaque section est pilotée par 5 paramètres sauf la section de sortie qui est fixée : on a donc 30 paramètres. Le cas n° 13 comporte 6 paramètres par section, ce qui fait en tout 63 degrés de liberté (sachant que les deux décentrement et le glissement de la section initiale sont bloqués).

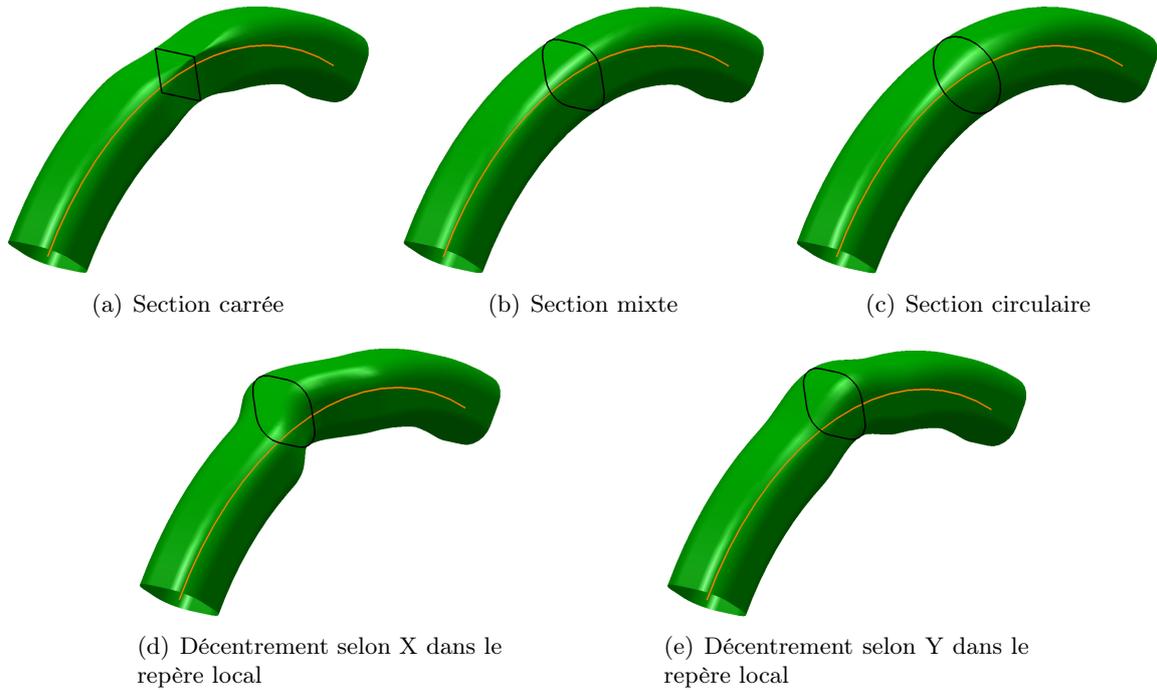


FIGURE 3.33 – Amélioration du paramétrage du coude avec 7 sections : les sections peuvent passer d'une forme circulaire à une forme rectangulaire de manière « continue ».

Cas	Nb de sections	Nb de paramètres	Déplacement initial [mm]	Déplacement optimisé [mm]	Nombre d'itérations	Taille de maille [mm]	Remarque
9	7	30	5,67	0,54	88	3	
10	7	30	5,68	0,55	66	2	
11	7	30	18,67	0,52	67	2	décentrement des sections 3 et 4 au départ
12	7	28	5,68	0,54	89	2,5	pas de décentrement pour la 1 ^{re} section
13	12	63	5,68	0,61	126	2,5	pas de décentrement pour la 1 ^{re} section

Tableau 3.5 – Minimisation du déplacement du nœud maître (cas n° 9 à 13)

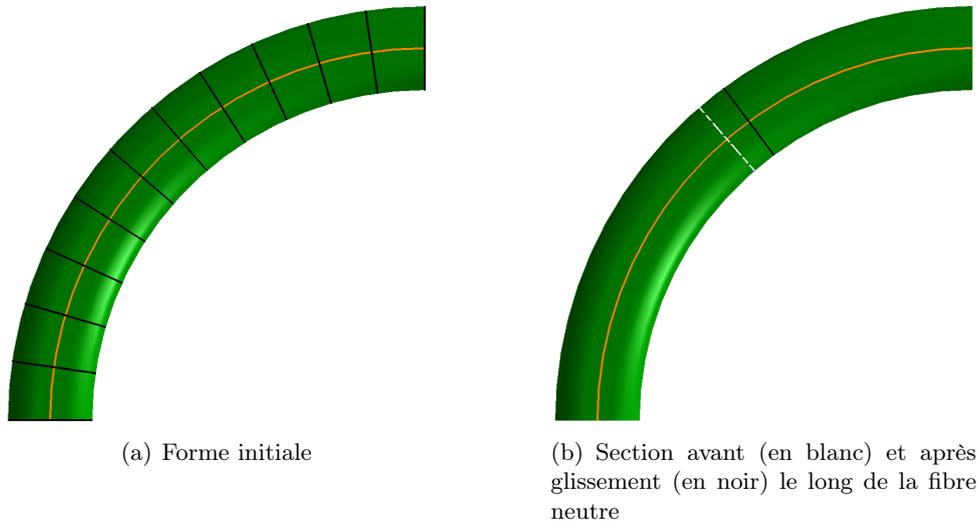


FIGURE 3.34 – Paramétrage du coude avec 12 sections : ces dernières peuvent glisser le long de la fibre neutre.

3.6 Discussion des résultats

En optimisation déterministe, un gradient faux met très vite un terme au processus d'optimisation car il propose des solutions aberrantes. Que nos algorithmes convergent à chaque fois nous rassure donc sur la justesse de notre démarche et témoigne également de la faisabilité de la méthode.

À propos du budget en essais, on observe que pour atteindre la convergence, il a fallu réaliser un nombre de calculs de l'ordre de une à trois fois le nombre de paramètres. Nous avons utilisé la méthode du gradient avec un pas constant qui est loin d'être l'algorithme le plus efficace. On observe d'ailleurs sur la figure 3.35(c) le comportement en zigzags souvent dénoncé pour cette méthode [Culioli 1994]. On note un assez gros potentiel d'amélioration en prenant par exemple un algorithme du gradient conjugué ou de quasi-Newton.

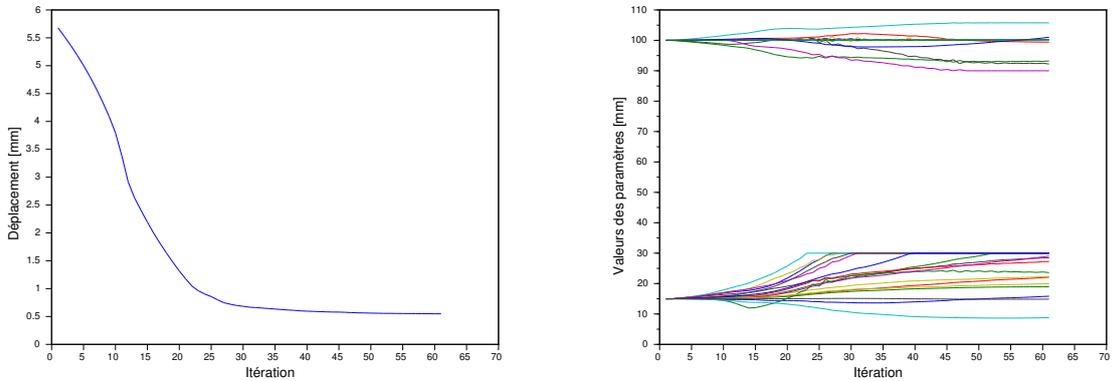
Néanmoins, avec une optimisation directe, chaque calcul est gagnant. Si le budget en essais est limité, on peut tout de suite améliorer la solution courante avec une méthode directe. Comparativement à une optimisation par plan d'expériences, on va rester proche de la solution courante mais avec moins d'essais, on va trouver plus vite de meilleures solutions. En mécanique des fluides, cela peut s'avérer très intéressant tant les calculs sont lourds. On peut tout à fait imaginer améliorer un design en lançant moins de 5 calculs.

Enfin, nous avons utilisé les gradients dans une optimisation directe mais il aurait été également intéressant de les utiliser pour construire une surface de réponse.

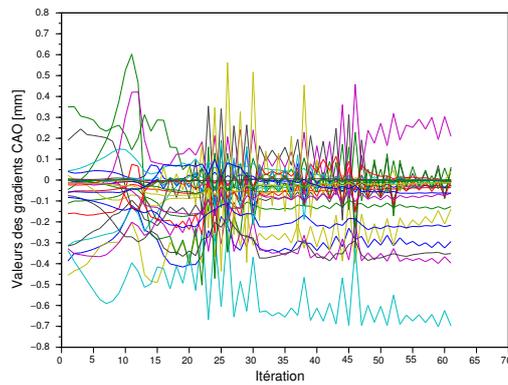
Extension de la méthode

Il reste un certain nombre de verrous à lever pour étendre la méthode :

- nous aimerions à l'avenir traiter des pièces au genre et au nombre de frontières hétéroclites (voir la figure 3.36). Or, pour un bon nombre de géométries, il sera nécessaire de les découper, soit sur des critères géométriques soit sur des critères topologiques;

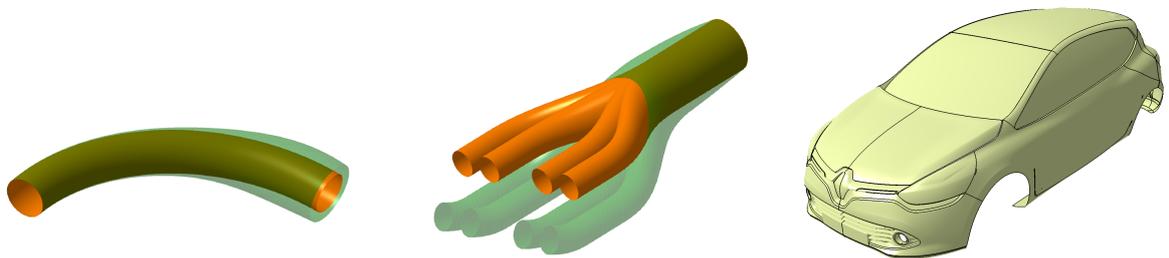


(a) Évolution du déplacement au fil des itérations (b) Évolution de la valeur des paramètres au fil des itérations



(c) Évolution des gradients CAO au fil des itérations

FIGURE 3.35 – Courbes des résultats du cas n° 10



(a) Coude : genre 0 - 2 frontières (b) Répartiteur : genre 0 - 5 frontières (c) Carrosserie : genre 0 - 1 frontière

FIGURE 3.36 – Diversité des cas d'étude

- o les perturbations appliquées aux paramètres sont définies de manière empirique. À l'avenir, il faudra déterminer leur amplitude de façon cohérente à la fois par rapport à l'influence du

- paramètre à perturber mais également par rapport à la précision du modeler géométrique et du mailleur ;
- il faudra trouver une méthode pour déterminer les points invariants des domaines paramétriques ;
 - on a constaté de fortes sensibilités par endroit lors du calcul des sensibilités du maillage. De même, les gradients issus des solveurs adjoints sont bruités. L'évaluation du gradient par rapport aux paramètres CAO semble lisser les gradients aux nœuds.
 - les contraintes liées à la conservation de certains objets géométriques n'ont pas été abordées dans cette partie. On parle par exemple des lignes caractéristiques liées à des contraintes de design, notamment pour toutes les pièces de carrosserie (lignes de carre, etc.). Il serait donc intéressant de se pencher sur le potentiel de la méthode à préserver les lignes caractéristiques entre les domaines paramétriques durant une évaluation ;
 - il sera nécessaire à l'avenir de prendre en compte dans le processus d'optimisation la présence de zones interdites. Le paramétrage initial étant fait dans un logiciel CAO, le solveur géométrique peut retourner des erreurs, c'est-à-dire que la forme souhaitée peut ne pas être générée.

Application à la mécanique des fluides

Les sensibilités fournies à l'heure actuelle par les logiciels de CFD ne sont pas correctes. Toutefois, il ne semble pas y avoir de limitation à l'utilisation de notre approche avec ces gradients. Du reste, nous avons mesuré la sensibilité du maillage d'un véhicule par rapport à un paramètre (cf. figure 3.37).

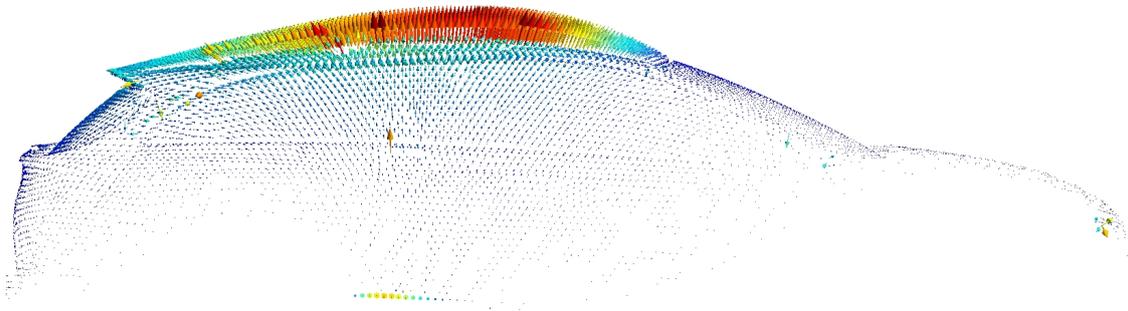


FIGURE 3.37 – Sensibilité du maillage par rapport à la flèche du pavillon

Bibliographie du 3^e chapitre

- [Armstrong 2007] C.G. Armstrong, T.T. Robinson, H. Ou et C. Othmer. *Linking adjoint sensitivity maps with CAD parameters*. Evolutionary methods for design, optimization and control, pages 234–239, 2007. (cité page 87)
- [Chen 1997] S. Chen et D.A. Tortorelli. *Three-dimensional shape optimization with variational geometry*. Structural optimization, vol. 13, no. 2-3, pages 81–94, 1997. (cité page 87)
- [Choi 1994] K.K. Choi et K.H. Chang. *A study of design velocity field computation for shape optimal design*. Finite Elements in Analysis and Design, vol. 15, no. 4, pages 317–341, 1994. (cité page 88)
- [Choi 2005] K.K. Choi et N.H. Kim. Structural sensitivity analysis and optimization 1. Springer New York, 2005. (cité page 88)
- [Culioli 1994] J.C. Culioli. Introduction à l’optimisation. Ellipses, 1994. (cité pages 26 et 115)
- [Floater 1997] M.S. Floater. *Parametrization and smooth approximation of surface triangulations*. Computer-Aided Geometric Design, vol. 14, no. 3, pages 231–250, 1997. (cité page 94)
- [Geuzaine 2009] C. Geuzaine et J.F. Remacle. *Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. International Journal for Numerical Methods in Engineering, vol. 79, no. 11, pages 1309–1331, 2009. (cité page 95)
- [Gu 2008] X.D. Gu et S.T. Yau. Computational conformal geometry, volume 3 of *Advanced lectures in mathematics*. International Press, 2008. (cité page 89)
- [Hardee 1999] E. Hardee, K.H. Chang, J. Tu, K.K. Choi, I. Grindeanu et X. Yu. *A CAD-based design parameterization for shape optimization of elastic solids*. Advances in Engineering Software, vol. 30, no. 3, pages 185–199, 1999. (cité page 87)
- [Kripac 1997] J. Kripac. *A mechanism for persistently naming topological entities in history-based parametric solid models*. Computer-Aided Design, vol. 29, no. 2, pages 113–122, 1997. (cité page 87)
- [Lindby 1999] T. Lindby et J.L.T. Santos. *Shape optimization of three-dimensional shell structures with the shape parametrization of a CAD system*. Structural optimization, vol. 18, no. 2-3, pages 126–133, 1999. (cité page 88)
- [Remacle 2010] J.F. Remacle, C. Geuzaine, G. Compère et E. Marchandise. *High-quality surface remeshing using harmonic maps*. International Journal for Numerical Methods in Engineering, vol. 83, pages 403–425, 2010. (cité pages 90, 91, 93, 95 et 98)
- [Robinson 2009] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer et T. Grahns. *Sensitivity-based optimization of parameterized CAD geometries*. In Proceedings of 8th World Congress on Structural and Multidisciplinary Optimization. ISSMO, 2009. (cité pages 87 et 89)
- [Robinson 2012] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer et T. Grahns. *Optimizing parameterized CAD geometries using sensitivities-based on adjoint functions*. Computer-Aided Design and Applications, pages 253–268, 2012. (cité pages xx et 89)
- [Toivanen 2006] J.I. Toivanen et J. Martikainen. *A new method for creating sparse design velocity fields*. Computer Methods in Applied Mechanics and Engineering, vol. 196, no. 1-3, pages 528–537, 2006. (cité page 88)

- [Yu 2011] G. Yu, J.D. Müller, D. Jones et F. Christakopoulos. *CAD-based shape optimisation using adjoint sensitivities*. *Computers & Fluids*, vol. 46, no. 1, pages 512–516, 2011. 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics. (cité page 87)

Bibliographie du 3^e chapitre

Conclusion et perspectives

Le processus de développement d'un produit est aujourd'hui porté essentiellement par des modèles CAO. Pour autant, ces derniers sont rarement le support sur lequel se font les études d'optimisation. Pour optimiser les épaisseurs ou les matériaux d'une caisse avec un cahier des charges en crash et en NVH, il est beaucoup plus aisé de travailler sur un maillage. En revanche, l'optimisation de forme en mécanique des fluides oblige à travailler avec un support CAO, notamment parce que ce dernier permet d'intégrer les paramètres métier et de prendre en compte des contraintes de fabrication au travers du paramétrage. Néanmoins, cela soulève par ailleurs deux difficultés : la grande dimension de l'espace des paramètres ainsi que le coût d'un calcul numérique.

Nous avons proposé dans ce manuscrit deux méthodes d'optimisation basées sur des modèles CAO. Nous nous sommes d'abord attachés à développer une méthode qui permet de passer d'une surface morte à un modèle paramétré. La frontière de l'espace de conception est modélisée par une approche statistique de type SVM.

Dans la première partie de cette thèse, nous avons suggéré deux approches pour réduire la dimension du problème : la première est une analyse en composantes principales basée sur une nouvelle description des géométries (voxellisation). Cette technique s'est avérée complexe à mettre en place. Une règle simple n'est pas applicable en l'état pour définir le nombre de degrés de liberté. Il s'avère également périlleux d'essayer de basculer de l'espace réduit à l'espace initial (paramètres CAO). La seconde approche a donné de bien meilleurs résultats. Elle utilise un réseau de neurones artificiels qui permet l'extraction de méta-paramètres. Par exemple, d'un problème à 20 paramètres, nous avons pu le réduire à 10 variables, divisant ainsi le nombre de degrés de liberté par deux. Pour contourner les calculs lourds, nous avons proposé une construction de la surface de réponse combinant deux niveaux de fidélité. Cela nous a donné un gain de temps d'un facteur 10.

Dans la seconde partie, nous avons adopté un tout autre scénario basé sur l'exploitation des gradients fournis par les solveurs adjoints de certains outils numériques. Cette deuxième méthode permet de lever très naturellement le verrou du nombre de paramètres. Une approche originale et efficace a été proposée pour étendre le calcul des gradients jusqu'aux paramètres CAO. Elle s'appuie sur des techniques de morphing à isoconnectivité. Le calcul des sensibilités est fait par différences finies. La comparaison a été rendue possible en projetant les formes nominale et perturbée sur une même forme géométrique canonique. Au travers d'exemples académiques, nous avons montré que notre approche est robuste et permet de résoudre les problèmes identifiés dans les méthodes recensées dans la littérature. Dans un contexte où le budget calcul est limité par la lourdeur des simulations, ce type d'approche représente un réel atout.

A l'issu des travaux, plusieurs perspectives peuvent être envisagées afin d'élargir le champ d'application de notre méthode. Elles concernent la gestion automatique du genre et du nombre de frontières permettant de prendre en compte plus de topologies. On peut aussi envisager

Conclusion et perspectives

de préciser le type de perturbation pouvant être appliqué à chaque paramètre, ainsi que la détermination des points invariants des domaines paramétriques. Enfin, on peut également citer la gestion des lignes caractéristiques et des zones interdites du domaine de conception. Notre approche permet d'envisager d'optimiser n'importe quel modèle CAO dès lors que les solveurs adjoints sont disponibles.

Quelques définitions

Sommaire

A.1 Continuité géométrique	123
---	------------

A.1 Continuité géométrique

Le terme de continuité paramétrique C^n a été introduit pour le distinguer de la continuité géométrique G^n . La continuité C^n , plus contraignante que G^n , traite des propriétés de raccord du paramétrage entre les patches alors que la continuité géométrique traite des propriétés de raccord entre les lieux géométriques associés. En d'autres termes, G^n supprime les restrictions sur la rapidité avec laquelle le paramètre retrace la courbe. Pour une continuité géométrique, n mesure la régularité de la courbe.

En dimension 2, soient deux patches : $\begin{cases} f : [0,1] \rightarrow \mathbb{R}^2 & C^\infty \text{ (polynomial)} \\ g : [0,1] \rightarrow \mathbb{R}^2 & C^\infty \text{ (polynomial)} \end{cases}$ qui se raccordent en $f(1) = g(0)$ (les courbes sont jointes : $C^0 \equiv G^0$).

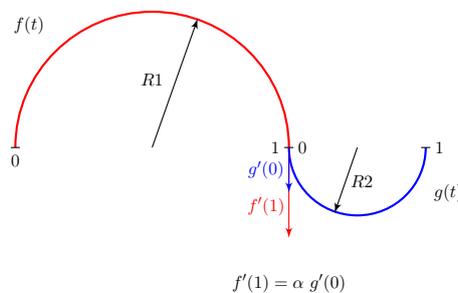


FIGURE A.1 – Continuité G^1

Annexe A. Quelques définitions

- Le raccord est dit C^1 si $f'(1) = g'(0)$;
- Le raccord est dit C^2 si on a la continuité C^1 et $f''(1) = g''(0)$;
- Le raccord est dit C^n si on a la continuité C^{n-1} et $f^{(n)}(1) = g^{(n)}(0)$.
- Le raccord est dit G^1 si $f'(1) = \alpha g'(0)$: les patches ont aussi une direction tangente commune au point de jonction ;
- Le raccord est dit G^2 si on a la continuité G^1 et $f''(1) = \beta g''(0)$: les patches ont aussi un rayon de courbure commun au point de jonction ;
- Le raccord est dit G^n si on a la continuité G^{n-1} et $f^{(n)}(1) = \gamma g^{(n)}(0)$: deux patches $f(t)$ et $g(t)$ ont une continuité G^n si la direction mais pas nécessairement la valeur des deux vecteurs est égale.

Par exemple, un rectangle dont l'un des coins serait arrondi a une continuité G^1 dans ce coin, mais pas de continuité G^2 . Pour des raisons esthétiques lors de la modélisation de certaines pièces, des niveaux élevés de continuité géométrique sont obligatoires. Par exemple, les réflexions dans une carrosserie de voiture n'apparaîtront pas lisses à moins que la surface ne présente une continuité G^2 .

Plans d'expériences

Sommaire

B.1 Plans d'expériences élémentaires	125
B.1.1 Plans de Monte-Carlo	125
B.1.2 Plans hypercubes latins	126
B.2 Plans d'expériences avancés	129
B.2.1 Substitution des points non faisables	129
B.2.2 Plans D-optimaux	129
B.2.3 Plans G-optimaux	130
B.2.4 Plans d'expériences auto-adaptatifs	130
B.3 Evaluation des performances d'un plan	135
B.3.1 Nombre de coefficients à estimer	135
B.3.2 Estimation des coefficients pour des modèles polynomiaux	135
B.3.3 Variance de prédiction	136
B.3.4 Critères de comparaison	137
Bibliographie des annexes	139

B.1 Plans d'expériences élémentaires

On note d la dimension de notre problème. Les plans ci-dessous permettent d'obtenir en sortie des plans d'expériences définis dans tout l'hypercube unité $[0,1]^d$:

- les plans de Monte-Carlo ;
- les plans hypercubes latins.

B.1.1 Plans de Monte-Carlo

Il s'agit de créer un plan d'expériences en choisissant des essais (n_{essais}) via un générateur pseudo-aléatoire (voir figure B.1).

Essais=rand(n_{essais},d)

Algorithme 1 : Monte-Carlo

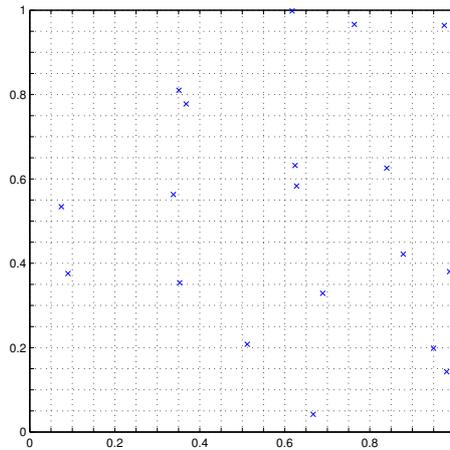


FIGURE B.1 – Plan de Monte-Carlo en dimension 2 avec 20 essais

B.1.2 Plans hypercubes latins

Pour chaque dimension, l'intervalle $[0,1]$ est découpé en autant de segments que d'essais puis un essai est disposé aléatoirement par segment (cf. figure B.2). La position des essais dans chaque segment peut être centrale ou aléatoire (on parle alors de plan LHS avec deux niveaux aléatoires). Pour améliorer la qualité du plan, on peut tirer successivement plusieurs plans et retenir le meilleur selon un critère, par exemple :

- en maximisant la distance minimale entre deux points de l'échantillon [Johnson 1990] ;
- en minimisant la corrélation entre les essais.

```
Essais(:,1)=transpose([0 : n_essais - 1] + 0.5)/n_essais
for i ← 2 to d do
    | Essais(:,i)=transpose(randperm(n_essais) - 0.5)/n_essais
end
```

Algorithme 2 : LHS avec 1 niveau aléatoire

```
Essais(:,1)=transpose([0 : n_essais - 1] + rand(1,n_essais))/n_essais
for i ← 2 to d do
    | Essais(:,i)=transpose(randperm(n_essais) - 1 + rand(1,n_essais))/n_essais
end
```

Algorithme 3 : LHS avec 2 niveaux aléatoires

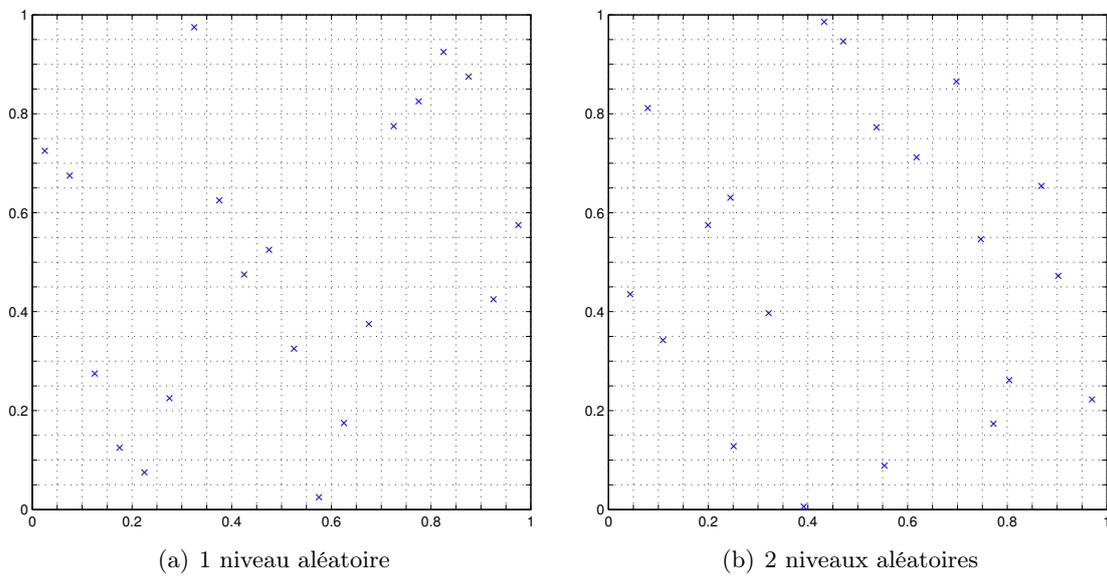


FIGURE B.2 – Plans LHS en dimension 2 avec 20 essais

B.2 Plans d'expériences avancés

Parmi tous les plans d'expériences qui existent, les plans orthogonaux sont les plus couramment employés dans l'industrie, notamment pour leur facilité de mise en oeuvre. Toutefois, ils exigent parfois un nombre d'essais trop élevé par rapport au budget calcul défini, et qui plus est, des contraintes de domaine peuvent interdire des essais et rendre impossible la construction de ce type de plan [Benoist 1994]. Ceci est notamment dû au fait que les essais sont situés sur les bords du domaine, là où se trouvent généralement des zones interdites. Bref, l'intérêt d'utiliser des tableaux orthogonaux n'est plus évident dès lors que le domaine de variation des paramètres n'est plus un hypercube.

Il est donc indispensable d'utiliser d'autres plans d'expériences qui sont optimaux par rapport à des critères statistiques. Les critères portent essentiellement sur la variance des prévisions que l'on cherchera (cf. [Benoist 1994]) :

- à minimiser pour obtenir une plus grande précision sur des estimations de combinaisons non-réalisées ;
- à rendre homogène sur tout le champ afin de pouvoir comparer les prévisions quelle que soit la combinaison considérée.

B.2.1 Substitution des points non faisables

Une idée simple est de remplacer chaque essai du plan initial qui a échoué par l'essai le plus proche issu d'une base de candidats.

B.2.2 Plans D-optimaux

Un plan D-optimal (D pour *déterminant maximum*) est la généralisation d'un plan orthogonal. Le principe du critère de D-optimalité consiste – pour un modèle mathématique postulé et en notant X la matrice du plan – à choisir parmi un grand nombre n_c d'essais candidats les n points expérimentaux qui donnent le déterminant de $X^t X$ le plus élevé. Maximiser ce déterminant permet de minimiser les éléments de $(X^t X)^{-1}$. Or, comme ses termes diagonaux sont proportionnels aux variances des coefficients, on obtient ainsi un plan qui minimise la variance globale de tous les coefficients du modèle.

En pratique, il n'est pas possible d'envisager de calculer tous les déterminants possibles et de choisir le plus élevé. Effectivement, le nombre de déterminants à calculer est égal au nombre de combinaisons de n_c objets pris n à n , *i.e.* :

$$C_n^{n_c} = \frac{n!}{(n - n_c)! n_c!} \tag{B.1}$$

Pour 20 essais choisis parmi 100, il faudrait calculer environ $5,36 \cdot 10^{20}$ déterminants... On utilise alors des algorithmes qui sont basés sur l'échange des points. On commence par définir un nombre élevé n_c de points candidats qui pourraient être inclus dans le plan final et qui tiennent compte des contraintes. On choisit ensuite un modèle mathématique a priori (linéaire, quadratique...). Enfin, on fixe le nombre n d'essais que l'on désire réaliser. L'algorithme cherche un déterminant élevé en opérant par échange. Il choisit au hasard n points (parmi les n_c points

candidats) qui forment le plan initial et il calcule le déterminant $X^t X$ correspondant. L'ordinateur retire un point des n points du plan initial et le remplace par un des points candidats. Il calcule le déterminant de la nouvelle matrice $X^t X$. Si ce deuxième déterminant est plus élevé que le premier, l'ordinateur conserve ce point, autrement il le rejette et prend un deuxième point parmi les points candidats. L'algorithme poursuit le calcul en échangeant tous les points du plan à n essais avec tous les points candidats. Le calcul se poursuit ainsi jusqu'à ce que la substitution d'un point par un autre ne fasse plus progresser suffisamment la valeur du déterminant [Goupy 1999].

B.2.3 Plans G-optimaux

Le G vient du critère anglais *General variance*. Les différents éléments de la diagonale principale de la matrice $(X^t X)^{-1}$ ne sont pas tous égaux entre eux. Les coefficients du modèle sont donc entachés d'erreurs plus ou moins fortes. Le critère de G-optimalité cherche à diminuer la valeur de la plus grande variance pour qu'elle soit la plus faible possible [Goupy 1999]. En pratique, un plan G-optimal est beaucoup plus lourd à calculer en terme de ressources informatiques. Il est à noter que les plans D- et G-optimaux sont équivalents sur des domaines cubiques.

B.2.4 Plans d'expériences auto-adaptatifs

Pour lancer nos études d'optimisation, nous avons besoin de connaître l'espace réel de conception de nos modèles CAO. Pour cela, nous avons utilisé les machines à vecteurs de support qui se basent sur un plan d'expériences (cf. partie 2.2.2 (p. 51)). En regard de cela, il arrive que l'on n'a pas lancé assez d'essais pour discriminer correctement la frontière. Il est donc opportun de compléter le plan d'expériences initial par de nouveaux essais. La méthode suivante s'inspire grandement des travaux de Marc Albertelli [Albertelli 2005].

Il faut construire la façon dont le plan d'expériences complète les essais de base. La première idée est de générer dans l'espace de conception un grand nombre de points et de ne retenir que les meilleurs en utilisant comme critère de sélection la distance euclidienne. Ce qui donne de bons résultats en 2D.

En notant :

- x_b : un essai de la base ;
- x_c : un essai candidat ;
- $p = 2$;
- d : dimension.

on a :

$$d_p(x_c, x_b) = \left(\sum_{k=1}^d |x_{c,k} - x_{b,k}|^p \right)^{\frac{1}{p}} \tag{B.2}$$

Seulement, plus la dimension augmente, plus les essais sélectionnés sont déportés vers les bords de l'hypercube. Ce phénomène d'attraction du vide peut se visualiser en affichant la répartition des essais (se reporter à la figure B.4).

Le nombre de coins de l'hypercube augmente avec le nombre de paramètres, de même que le nombre de bords (cf. tableau B.1).

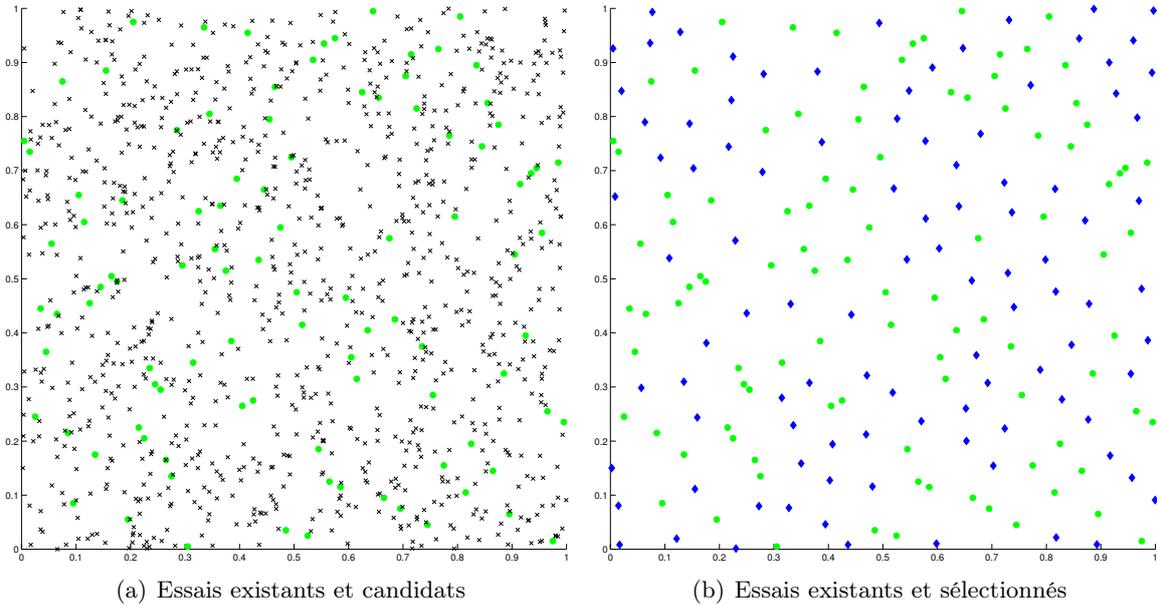


FIGURE B.3 – Complètement d'un plan LHS en 2D (100 essais existants ●, 1000 essais candidats x, 100 essais sélectionnés ◆)

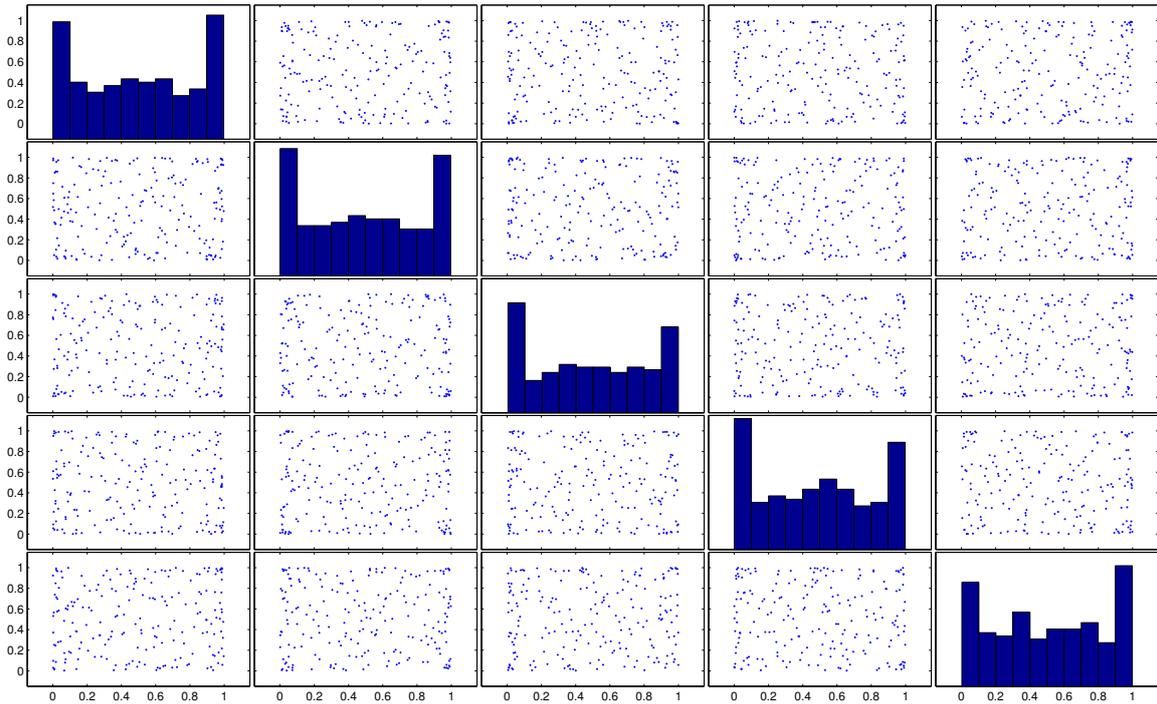


FIGURE B.4 – Répartition des essais en 5D (100 essais existants, 10000 essais candidats, 100 essais sélectionnés)

Afin de casser l'effet du vide, les essais de la base sont dupliqués dans toutes les dimensions. La base est donc augmentée de $(3^d - 1)$ essais. La sélection des essais à planifier se fait toujours

Annexe B. Plans d'expériences

Nombre de paramètres	Nombre de coins	Bords
1	2	2 points
2	4	4 droites
3	8	6 surfaces
4	16	?
...	...	?
10	1024	?
...	...	?
n	2^n	?

Tableau B.1 – Evolution du nombre de coins et de bords avec la dimension

parmi les essais candidats mais en tenant compte des essais originaux et de leurs clones.

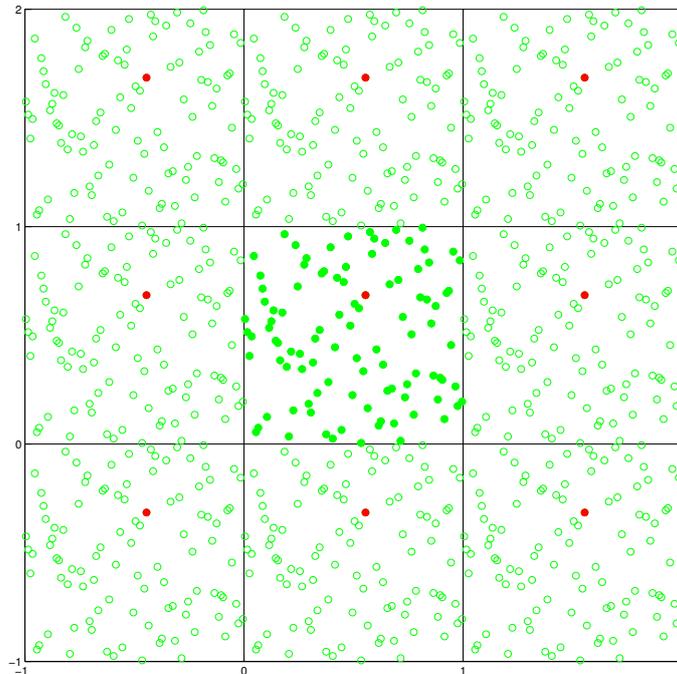


FIGURE B.5 – Duplication des essais dans l'hypercube $[-1,2]^2$. En rouge sont représentés un essai original de la base et ses clones.

Toutefois, le nombre de distances à calculer augmente considérablement ($3^d - 1$ distances en plus). Comme la sélection finale des essais repose sur la distance minimale entre un essai candidat et tous les essais de la base, nous allons introduire une nouvelle distance qui permet de simplifier les calculs.

Soient x_c un essai candidat, x_b un essai original de la base, $x_b(\Delta)$ l'ensemble de ses clones et $x_b^v = \{x_b, x_b(\Delta)\}$. Soit $\tilde{d}_p(x_c, x_b) = \min_{x \in x_b^v} (d_p(x_c, x))^p$ la distance minimale entre x_c et l'ensemble

x_b^v . Cette distance est donnée par l'équation (B.3) :

$$\tilde{d}_p(x_c, x_b) = \sum_{k=1}^d \min(|x_{c,k} - x_{b,k}|, 1 - |x_{c,k} - x_{b,k}|)^p \quad (\text{B.3})$$

En effet, l'ensemble des clones est $x_b(\Delta) = (x_{b,1} + \Delta_1, x_{b,2} + \Delta_2, \dots, x_{b,d} + \Delta_d)$ avec $\Delta \in \{-1, 0, 1\}^d$. Il vient donc :

$$\begin{aligned} \tilde{d}_p(x_c, x_b) &= \min_{\Delta \in \{-1, 0, 1\}^d} d_p(x_c, x_b(\Delta))^p \\ &= \min_{\Delta \in \{-1, 0, 1\}^d} \left(\underbrace{\sum_{k=1}^d |x_{c,k} - x_{b,k} - \Delta_k|^p}_{\text{les termes de la somme sont indépendants}} \right) \\ &= \sum_{k=1}^d \min_{\Delta \in \{-1, 0, 1\}^d} |x_{c,k} - x_{b,k} - \Delta_k|^p \\ &= \sum_{k=1}^d \min(|x_{c,k} - x_{b,k}|, |x_{c,k} - x_{b,k} + 1|, |x_{c,k} - x_{b,k} - 1|)^p \\ &= \sum_{k=1}^d \min(|x_{c,k} - x_{b,k}|, 1 - |x_{c,k} - x_{b,k}|, 1 + |x_{c,k} - x_{b,k}|)^p \\ &= \sum_{k=1}^d \min(|x_{c,k} - x_{b,k}|, 1 - |x_{c,k} - x_{b,k}|)^p \end{aligned}$$

Ainsi, on obtient une répartition des essais uniforme, comme on peut le voir sur la figure B.6.

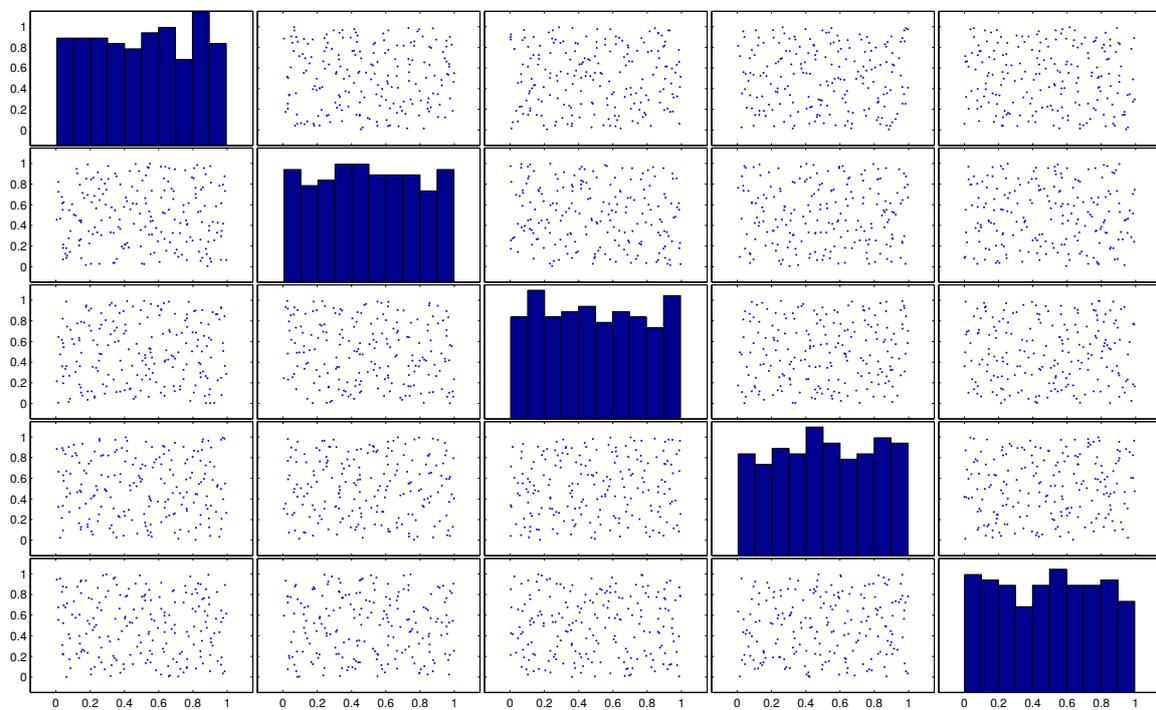


FIGURE B.6 – Répartition des essais en 5D avec la nouvelle distance (100 essais existants, 1000 essais candidats, 100 essais sélectionnés, $p=1$)

B.3 Evaluation des performances d'un plan

B.3.1 Nombre de coefficients à estimer

L'ingénieur définit un modèle mathématique *a priori* le plus proche possible du modèle réel. Les modèles polynomiaux sont les plus employés et sont la plupart du temps du premier ou du deuxième degré :

$$y = \beta_0 + \sum_{i=1}^n \beta_i x_i + \sum_{i=1}^n \beta_{ii} x_i^2 + \sum_{i < j}^n \beta_{ij} x_i x_j \quad (\text{B.4})$$

Le nombre de coefficients à estimer dépend du modèle retenu :

$$\underbrace{\underbrace{1}_{\text{constante}} + \underbrace{2}_{\text{termes quadratiques}} + \underbrace{n}_{\text{termes linéaires}}}_{\text{modèle quadratique pur}} + \underbrace{\frac{n \cdot (n-1)}{2}}_{\text{interactions}} \quad (\text{B.5})$$

modèle quadratique

ce qui donne :

Modèle	Nombre de paramètres												
	1	2	3	4	5	6	7	8	9	10	15	20	25
linéaire	2	3	4	5	6	7	8	9	10	11	16	21	26
linéaire avec interactions	2	4	7	11	16	22	29	37	46	56	121	211	326
quadratique pur	3	5	7	9	11	13	15	17	19	21	31	41	51
quadratique	3	6	10	15	21	28	36	45	55	66	136	231	351

Tableau B.2 – Nombre de coefficients à estimer

Si l'on prend le cas d'un problème en dimension 20 approché avec un modèle quadratique, il faudrait 231 essais pour estimer tous les coefficients du modèle.

B.3.2 Estimation des coefficients pour des modèles polynomiaux

Dans le cas des modèles polynomiaux, le calcul des coefficients (constants) β pour des modèles de régression linéaires est réalisé selon la méthode des moindres carrés :

$$y = X\beta + \epsilon \quad (\text{B.6})$$

en notant X la matrice du plan pour le modèle choisi. On souhaite trouver le vecteur des estimateurs $\hat{\beta}$ au sens des moindres carrés qui minimise :

$$L = \sum_{j=1}^n \epsilon_j^2 = \epsilon^t \epsilon = (y - X\beta)^t (y - X\beta) \quad (\text{B.7})$$

Il vient :

$$\begin{aligned}
 L &= y^t y - \underbrace{\beta^t X^t y}_{\text{scalaire}} - \underbrace{y^t X \beta}_{(\beta^t X^t y)^t} + \beta^t X^t X \beta \\
 &= y^t y - 2 \cdot \beta^t X^t y + \beta^t X^t X \beta
 \end{aligned}
 \tag{B.8}$$

Les paramètres optimaux au sens des moindres carrés doivent vérifier :

$$\left. \frac{\partial L}{\partial \beta} \right| = -2X^t y + 2X^t X \hat{\beta} = 0
 \tag{B.9}$$

d'où :

$$X^t X \hat{\beta} = X^t y
 \tag{B.10}$$

L'estimateur au sens des moindres carrés de β est :

$$\hat{\beta} = (X^t X)^{-1} X^t y
 \tag{B.11}$$

Il a été montré par les statisticiens (on retrouve la démonstration dans [Montgomery 2005]) que la matrice des variances-covariances des coefficients vaut :

$$Var(\hat{\beta}) = \sigma^2 (X^t X)^{-1}
 \tag{B.12}$$

avec σ^2 la variance des résidus (également appelée *variance de régression*).

Les variances des coefficients sont disposées sur la diagonale principale de $Var(\hat{\beta})$ et les covariances sont les éléments non diagonaux. Dans le calcul des coefficients, il y a trois composantes qui induisent des erreurs :

- l'erreur commise sur les réponses ;
- l'emplacement des points expérimentaux ;
- le modèle mathématique choisi *a priori*.

B.3.3 Variance de prédiction

Soit un point P du domaine d'étude. La réponse en ce point est donnée par :

$$\hat{y}_p = x_p^t \hat{\beta}
 \tag{B.13}$$

avec $x_p^t = [1, x_1, \dots, x_d, x_1^2, \dots, x_d^2, x_1 \cdot x_2, \dots]$ une fonction des coordonnées du point P dépendante du modèle retenu. L'incertitude sur les coefficients se transmet à la réponse estimée, ce qui nous donne la formule de la variance de prédiction [Myers 1992] :

$$\begin{aligned}
 Var[\hat{y}_p(x_p)] &= Var(x_p^t \hat{\beta}) \\
 &= x_p^t Var(\hat{\beta}) x_p \\
 &= \sigma^2 x_p^t (X^t X)^{-1} x_p
 \end{aligned}
 \tag{B.14}$$

Les erreurs sur les réponses prédites proviennent donc de l'erreur commise sur les réponses mesurées, de l'emplacement des points expérimentaux dans le domaine d'étude, du modèle mathématique choisi *a priori* et de la position du point dont on calcule la réponse prédite.

La variance de prédiction échelonnée est estimée par :

$$v(x_p) = \frac{n \cdot \text{Var} [\hat{y}_p(x_p)]}{\sigma^2} = n \cdot x_p^t (X^t X)^{-1} x_p \quad (\text{B.15})$$

Elle ne dépend pas de la variable de réponse mais de la matrice du plan X . En d'autres termes, la variance de prédiction échelonnée est fonction du nombre d'essais n , de la matrice de covariance et du point de calcul. Il est donc possible d'évaluer la qualité de l'expérimentation avant de réaliser les essais.

B.3.4 Critères de comparaison

Afin d'évaluer la performance des plans, on retient trois critères :

- maximisation de la distance minimale entre deux points ;
- minimisation du maximum de la variance de dispersion échelonnée ;
- maximisation du déterminant de $X^t X$.

La figure B.7 présente une comparaison entre quatre types de plan d'expériences en dimension 20 avec un modèle quadratique.

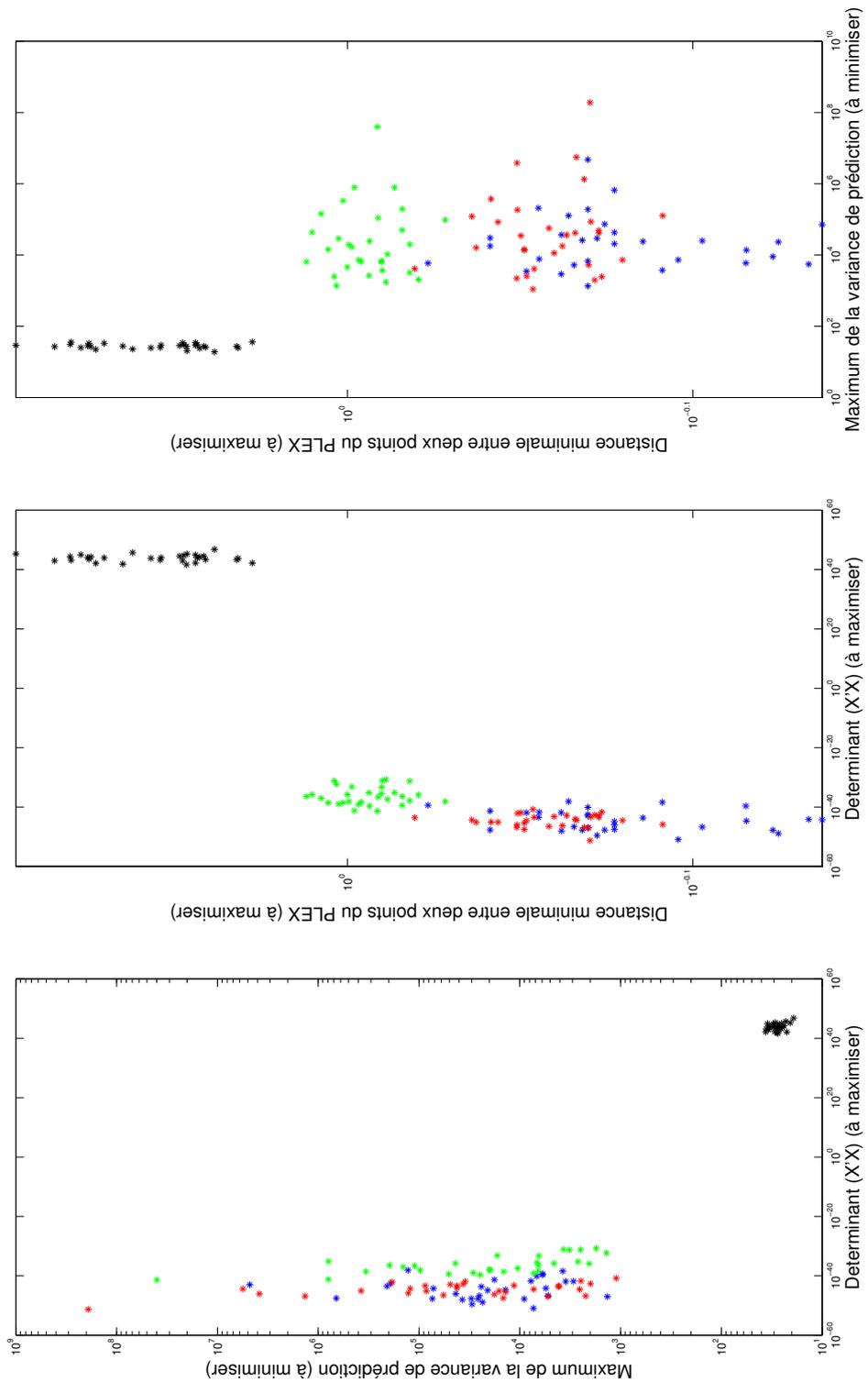


FIGURE B.7 – Comparaison des différents types de plan d'expériences : plans de Monte-Carlo *, plans LHS *, plans auto-adaptatifs *, plans D-optimaux * (dimension 20 avec une zone interdite (taux de reconstruction de 85,70 %), 30 points par type de plan, 10000 essais de validation, modèle quadratique)

Bibliographie des annexes

- [Albertelli 2005] M. Albertelli. *Projet MAPAO, plan d'expériences auto-adaptatif*. Rapport technique, Renault SA, 2005. (cité pages 52, 53, 54 et 130)
- [Benoist 1994] D. Benoist, Y. Tourbier et S. Germain-Tourbier. *Plans d'expériences : construction et analyse*. Technique & Documentation-Lavoisier, 1994. (cité pages 18, 19 et 129)
- [Goupy 1999] J. Goupy. *Plans d'expériences pour surfaces de réponse*. Dunod, 1999. (cité pages 19 et 130)
- [Johnson 1990] M.E. Johnson, L.M. Moore et D. Ylvisaker. *Minimax and maximin distance designs*. *Journal of Statistical Planning and Inference*, vol. 26, no. 2, pages 131–148, 1990. (cité page 126)
- [Montgomery 2005] D.C. Montgomery. *Design and analysis of experiments*. JohnWiley & Sons, 6^e édition, 2005. (cité page 136)
- [Myers 1992] R.H. Myers, G. Vining, A. Giovannitti-Jensen et S.L. Myers. *Variance dispersion properties of second order response surface designs*. *Journal of Quality Technology*, vol. 24, pages 1–11, 1992. (cité page 136)

Bibliographie globale

- [Albertelli 2005] M. Albertelli. *Projet MAPAO, plan d'expériences auto-adaptatif*. Rapport technique, Renault SA, 2005. (cité pages 52, 53, 54 et 130)
- [Allaire 2003] G. Allaire, F. Jouve et A.M. Toader. *Structural optimization using sensitivity analysis and a level-set method*, 2003. (cité page 17)
- [Allaire 2004] G. Allaire et F. Jouve. *A level-set method for vibration and multiple loads structural optimization*, 2004. (cité page 17)
- [Allaire 2007] G. Allaire. *Conception optimale de structures*. Springer Berlin Heidelberg, 2007. (cité pages 16 et 26)
- [Anderson 1999] W.K. Anderson et V. Venkatakrisnan. *Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation*. *Computers and Fluids*, vol. 28, page 443–480, 1999. (cité page 29)
- [Anderson 2003] W.K. Anderson et V. Venkatakrisnan. *Topology optimization of fluids in Stokes flow*. *Int. J. Num. Meth. Fluids*, vol. 41, page 77, 2003. (cité page 30)
- [Armstrong 2007] C.G. Armstrong, T.T. Robinson, H. Ou et C. Othmer. *Linking adjoint sensitivity maps with CAD parameters*. *Evolutionary methods for design, optimization and control*, pages 234–239, 2007. (cité page 87)
- [Aubert 2010] S. Aubert, P. Ferrand, F. Pacull et M. Buisson. *Fast CFD for shape and flow parameterization with meta-models built on high-order derivatives - Applications to fast design*. 27th International Congress of the Aeronautical Science, 2010. (cité page 25)
- [Bendsoe 2003] M.P. Bendsoe et O. Sigmund. *Topology optimization : theory, methods and applications*. Springer Berlin Heidelberg, 2003. (cité page 12)
- [Benoist 1994] D. Benoist, Y. Tourbier et S. Germain-Tourbier. *Plans d'expériences : construction et analyse*. Technique & Documentation-Lavoisier, 1994. (cité pages 18, 19 et 129)
- [Binois 2013] M. Binois. *Synthèse sur l'optimisation multi-objectifs*. Rapport technique, Consortium ReDICE, 2013. (cité page 23)
- [Blazek 2005] J. Blazek. *Computational fluid dynamics : principles and applications*. Elsevier, 2^e édition, 2005. (cité page 78)
- [Boser 1992] B.E. Boser, I.M. Guyon et V.N. Vapnik. *A training algorithm for optimal margin classifiers*. 1992. (cité page 52)
- [Bourlard 1988] H. Bourlard et Y. Kamp. *Auto-association by multilayer perceptrons and singular value decomposition*. *Biological Cybernetics*, vol. 59, pages 291–294, 1988. (cité page 68)
- [Boyd 2004] S. Boyd et L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. (cité page 26)

- [Brujic 2010] D. Brujic, M. Ristic, M. Mattone, P. Maggiore et G.P. De Poli. *CAD based shape optimization for gas turbine component design*. Structural and Multidisciplinary Optimization, vol. 41, pages 647–659, 2010. (cité page 10)
- [Chatterjee 2000] A. Chatterjee. *An introduction to the proper orthogonal decomposition*. Current Science, vol. 78, no. 7, 2000. (cité page 59)
- [Chen 1997] S. Chen et D.A. Tortorelli. *Three-dimensional shape optimization with variational geometry*. Structural optimization, vol. 13, no. 2-3, pages 81–94, 1997. (cité page 87)
- [Choi 1994] K.K. Choi et K.H. Chang. *A study of design velocity field computation for shape optimal design*. Finite Elements in Analysis and Design, vol. 15, no. 4, pages 317–341, 1994. (cité page 88)
- [Choi 2005] K.K. Choi et N.H. Kim. Structural sensitivity analysis and optimization 1. Springer New York, 2005. (cité page 88)
- [Collobert 2000] R. Collobert. *Support vector machines : théorie et application*. Rapport technique, IDIAP, 2000. (cité page 55)
- [Cottrell 2009] J.A. Cottrell, T.J.R. Hughes et Y. Bazilevs. Isogeometric analysis, toward integration of CAD and FEA. John Wiley and Sons, 2009. (cité page 12)
- [Cristianini 2000] N. Cristianini et J. Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, 2000. (cité page 52)
- [Culioli 1994] J.C. Culioli. Introduction à l’optimisation. Ellipses, 1994. (cité pages 26 et 115)
- [Dapogny 2013] C. Dapogny. *Shape optimization, level set methods on unstructured meshes and mesh evolution*. PhD thesis, UPMC, 2013. (cité pages 13 et 14)
- [de Falco 2011] C. de Falco, A. Reali et R. Vázquez. *GeoPDEs : a research tool for isogeometric analysis of PDEs*. Advances in Engineering Software, vol. 42, no. 12, pages 1020–1034, 2011. (cité page 12)
- [Devalan 2009] P. Devalan. *Simulation numérique dans le processus de conception de systèmes mécaniques*. Techniques de l’ingénieur, no. bm5013, 2009. (cité page 5)
- [Dhatt 2007] G. Dhatt, G. Touzot et E. Lefrançois. Méthode des éléments finis. Hermes, 2007. (cité page 41)
- [du Cauzé de Nazelle 2013] P. du Cauzé de Nazelle. *Paramétrage de formes surfaciques pour l’optimisation*. PhD thesis, École Centrale de Lyon, 2013. (cité page 12)
- [Dupuy 2007] D. Dupuy et C. Helbert. *Retour d’expériences sur métamodèles : partie théorique*. Rapport technique, Consortium DICE, 2007. (cité page 21)
- [Fletcher 1980] R. Fletcher. Practical methods of optimization. John Wiley & Sons, 1980. (cité page 26)
- [Floater 1997] M.S. Floater. *Parametrization and smooth approximation of surface triangulations*. Computer-Aided Geometric Design, vol. 14, no. 3, pages 231–250, 1997. (cité page 94)
- [Frey 1999] P.J. Frey et P.L. George. Maillages : applications aux éléments finis. Hermes, 1999. (cité page 42)
- [Friedman 1991] J.H. Friedman. *Multivariate Adaptive Regression Splines*. The annale of statistics, vol. 19, pages 1–141, 1991. (cité page 21)

- [Froment 2012] P. Froment, F. Gillot, L. Jézéquel et Y. Tourbier. *CAD-based multi-objective shape optimization method with meta-parameters*. Proceedings of the 10th world congress on computational mechanics, 2012. (cité page 80)
- [Gardan 2002] Y. Gardan. *CAO : modélisation géométrique*. Techniques de l'ingénieur, no. h3750, 2002. (cité page 44)
- [George 1991] P.L. George. Génération automatique de maillages : applications aux méthodes d'éléments finis. Masson, 1991. (cité pages 41 et 42)
- [Geuzaine 2009] C. Geuzaine et J.F. Remacle. *Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. International Journal for Numerical Methods in Engineering, vol. 79, no. 11, pages 1309–1331, 2009. (cité page 95)
- [Ginsbourger 2009] D. Ginsbourger, D. Dupuy, A. Badaea, L. Carraro et O. Roustant. *A note on the choice and the estimation of kriging models for the analysis of deterministic computer experiments*. Applied Stochastic Models in Business and Industry, vol. 25, pages 115–131, 2009. (cité page 22)
- [Goupy 1999] J. Goupy. Plans d'expériences pour surfaces de réponse. Dunod, 1999. (cité pages 19 et 130)
- [Grondin 2005] G. Grondin, V. Kelner, P. Ferrand et S. Moreau. *Robust design and parametric performance study of an automotive fan blade by coupling multi-objective genetic optimization and flow parameterization*. Proceedings of the International Congress of Fluid Dynamics Applications in Ground Transportation, 2005. (cité page 24)
- [Gu 2008] X.D. Gu et S.T. Yau. Computational conformal geometry, volume 3 of *Advanced lectures in mathematics*. International Press, 2008. (cité page 89)
- [Gunn 1998] S.R. Gunn. *Support vector machines for classification and regression*, 1998. (cité page 52)
- [Hardee 1999] E. Hardee, K.H. Chang, J. Tu, K.K. Choi, I. Grindeanu et X. Yu. *A CAD-based design parameterization for shape optimization of elastic solids*. Advances in Engineering Software, vol. 30, no. 3, pages 185–199, 1999. (cité page 87)
- [Haskins 2007] C. Haskins. Systems engineering handbook : a guide for system life cycle processes and activities. Numéro INCOSE-TP-2003-002-03.1. International Council on Systems Engineering, 2007. (cité page 4)
- [Hayes 2006] J.C. Hayes, P.G. Doctor, T.R. Heimbigner, C.W. Hubbard, L.J. Kangas, P.E. Keller, J.I. McIntyre, B.T. Schrom et R. Suarez. *Application of artificial neural network modeling to the analysis of the automated radioxenon sampler-analyser state of health sensors*. Proceedings of 28th Seismic Research Review, pages 793–800, 2006. (cité page 68)
- [Hucho 1998] W.H. Hucho. Aerodynamics of road vehicles. SAE International, 4^e édition, 1998. (cité page 72)
- [Hughes 2005] T.J.R. Hughes, J.A. Cottrell et Y. Bazilevs. *Isogeometric analysis : CAD, finite elements, NURBS, exact geometry and mesh refinement*. Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 39-41, pages 4135–4195, 2005. (cité pages 11 et 12)
- [Jacobs 1991] R.A. Jacobs, M.I. Jordan, S.J. Nowlan et G.E. Hinton. Adaptive mixture of local experts, volume 3. 1991. (cité pages 20 et 22)

- [Jameson 1988] A. Jameson. *Aerodynamic design via control theory*. Journal of Scientific Computing, vol. 3, pages 233–260, 1988. (cité page 27)
- [Jameson 1994] A. Jameson et J. Reuther. *Control theory based airfoil design using the Euler equations*. AIAA, 1994. (cité page 27)
- [Jameson 1995] A. Jameson. *Optimum aerodynamic design using CFD and control theory*. AIAA, 1995. (cité page 27)
- [Jameson 1998] A. Jameson, N. Pierce et L. Martinelli. *Optimum aerodynamic design using the Navier-Stokes equations*. AIAA, 1998. (cité page 27)
- [Johnson 1990] M.E. Johnson, L.M. Moore et D. Ylvisaker. *Minimax and maximin distance designs*. Journal of Statistical Planning and Inference, vol. 26, no. 2, pages 131–148, 1990. (cité page 126)
- [Kooperberg 1997] C. Kooperberg, S. Bose et C.J. Stone. *Polychotomous regression*. Journal of the American statistical association, vol. 92, pages 117–127, 1997. (cité page 21)
- [Kripac 1997] J. Kripac. *A mechanism for persistently naming topological entities in history-based parametric solid models*. Computer-Aided Design, vol. 29, no. 2, pages 113–122, 1997. (cité page 87)
- [Lebart 1982] L. Lebart, A. Morineau et J.-P. F  nelon. *Traitement des donn  es statistiques*. Dunod, 2^e   dition, 1982. (cité page 60)
- [Lebrun 2003] M. Lebrun. *Simulation et CAO en automatique et m  catronique*. Techniques de l’ing  nieur, no. s7260, 2003. (cité page 4)
- [Lindby 1999] T. Lindby et J.L.T. Santos. *Shape optimization of three-dimensional shell structures with the shape parametrization of a CAD system*. Structural optimization, vol. 18, no. 2-3, pages 126–133, 1999. (cité page 88)
- [Luzeaux 2011] D. Luzeaux, J.R. Ruault et J.L. Wippler. *Ma  trise de l’ing  nierie des syst  mes complexes et des syst  mes de syst  mes :   tude de cas*. Herm  s science publications-Lavoisier, 2011. (cité page 4)
- [Matheron 1963] G. Matheron. *Trait   de g  ostatistique appliqu  e, 2 : le krigeage*. Technip, 1963. (cité page 21)
- [Michailidis 2014] G. Michailidis. *Manufacturing constraints and multi-phase shape and topology optimization via a level-set method*. PhD thesis,   cole Polytechnique, 2014. (cité page 13)
- [Montgomery 2005] D.C. Montgomery. *Design and analysis of experiments*. JohnWiley & Sons, 6^e   dition, 2005. (cité page 136)
- [Moos 2004] O. Moos, F. Klimetzek et R. Rossmann. *Bionic optimization of air-guiding systems*. SAE Technical Paper, 2004. (cité pages 13 et 15)
- [Myers 1992] R.H. Myers, G. Vining, A. Giovannitti-Jensen et S.L. Myers. *Variance dispersion properties of second order response surface designs*. Journal of Quality Technology, vol. 24, pages 1–11, 1992. (cité page 136)
- [Nocedal 2006] J. Nocedal et S. Wright. *Numerical optimization*. John Wiley & Sons, 2006. (cité page 26)
- [Othmer 2005] C. Othmer et Th. Grahs. *Approaches to fluid dynamic optimization in the car development process*. EUROGEN, 2005. (cité page 27)

- [Peter 2006] J. Peter, S. Burguburu et M. Marcelet. Introduction à l'optimisation de forme en aérodynamique et quelques exemples d'application. *ONERA*, 2006. (cité page 29)
- [Pironneau 1984] O. Pironneau. Optimal shape design for elliptic systems. Springer-Verlag, 1984. (cité page 27)
- [Raghavan 2012] B. Raghavan. *Towards simultaneous meta-modeling for both the output and input spaces in the context of design shape optimization using asynchronous high-performance computing*. PhD thesis, Université de Technologie de Compiègne, 2012. (cité page 64)
- [Raghavan 2013] B. Raghavan, P. Breitkopf, Y. Tourbier et P. Villon. *Towards a space reduction approach for efficient structural shape optimization*. Structural and Multidisciplinary Optimization, vol. 48, no. 5, pages 987–1000, 2013. (cité pages 59 et 64)
- [Remacle 2010] J.F. Remacle, C. Geuzaine, G. Compère et E. Marchandise. *High-quality surface remeshing using harmonic maps*. International Journal for Numerical Methods in Engineering, vol. 83, pages 403–425, 2010. (cité pages 90, 91, 93, 95 et 98)
- [Robinson 2009] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer et T. Grahs. *Sensitivity-based optimization of parameterized CAD geometries*. In Proceedings of 8th World Congress on Structural and Multidisciplinary Optimization. ISSMO, 2009. (cité pages 87 et 89)
- [Robinson 2012] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer et T. Grahs. *Optimizing parameterized CAD geometries using sensitivities-based on adjoint functions*. Computer-Aided Design and Applications, pages 253–268, 2012. (cité pages xx et 89)
- [Roy 2008] R. Roy, S. Hinduja et R. Teti. *Recent advances in engineering design optimisation : challenges and future trends*. CIRP Annals - Manufacturing Technology, vol. 57, pages 697–715, 2008. (cité pages 8 et 10)
- [Saitou 2005] K. Saitou, K. Izui, S. Nishiwaki et P. Papalambros. *A survey of structural optimization in mechanical product development*. Journal of Computing and Information Science in Engineering, vol. 5, pages 214–226, 2005. (cité pages 8 et 10)
- [Scholz 2007] M. Scholz, M. Fraunholz et J. Selbig. Principal manifolds for data visualization and dimension reduction, volume 58. Springer Berlin Heidelberg, 2007. (cité page 66)
- [Scholz 2012] M. Scholz. *Validation of nonlinear PCA*. Neural Processing Letters, vol. 36, no. 1, pages 21–30, 2012. (cité page 68)
- [Spalart 1997] P.R. Spalart, W.H. Jou, M. Stretlets et S.R. Allmaras. *Comments on the feasibility of LES for wings and on the hybrid RANS/LES approach*. Proceedings of the 1st AFOSR International Conference on DNS/LES, 1997. (cité page 70)
- [Staten 2011] M.L. Staten, S.J. Owen, S.M. Shontz, A.G. Salinger et T.S. Coffey. *A comparison of mesh morphing methods for 3D shape optimization*. Proceedings of the 20th international meshing round-table, pages 293–311, 2011. (cité page 11)
- [Suh 2002] M.W. Suh, J.H. Lee, K.Y. Cho et S.I. Kim. *Section property method and section shape method for the optimum design of vehicle body structures*. International Journal of Vehicle Design, vol. 30, pages 115–134, 2002. (cité page 9)
- [Thévenin 2008] D. Thévenin et G. Janiga. Optimization and computational fluid dynamics. Springer, 2008. (cité page 27)

Bibliographie globale

- [Toivanen 2006] J.I. Toivanen et J. Martikainen. *A new method for creating sparse design velocity fields*. Computer Methods in Applied Mechanics and Engineering, vol. 196, no. 1-3, pages 528–537, 2006. (cité page 88)
- [Torres 2003] R. Torres, R. Salas, H. Allende et C. Moraga. *Robust expectation maximization learning algorithm for mixture of experts*. Computational Methods in Neural Modeling, vol. 2686, pages 238–245, 2003. (cité page 22)
- [van der Maaten 2009] L.J.P. van der Maaten, E.O. Postma et H.J. van den Herik. *Dimensionality reduction : a comparative review*. Rapport technique, Tilburg University, 2009. (cité page 66)
- [Vyshinsky 2010] V.V. Vyshinsky, Y.A. Dorofeev et Y.N. Sviridenko. *Fast aerodynamic design technologies*. In 27th International Congress of the Aeronautical Sciences, 2010. (cité page 67)
- [Yang 1992] R.J. Yang, A. Lee et D.T. McGeen. *Application of basis function concept to practical shape optimization problems*. Structural optimization, vol. 5, pages 55–63, 1992. (cité page 11)
- [Yu 2011] G. Yu, J.D. Müller, D. Jones et F. Christakopoulos. *CAD-based shape optimisation using adjoint sensitivities*. Computers & Fluids, vol. 46, no. 1, pages 512–516, 2011. 10th ICFD Conference Series on Numerical Methods for Fluid Dynamics. (cité page 87)

Liste des acronymes et abréviations

<i>i.e.</i>	<i>id est.</i>
<i>s.t.</i>	<i>subject to.</i>
ACP	analyse en composantes principales.
AFOSR	air force office of scientific research.
AIAA	American institute of aeronautics and astronautics.
API	application programming interface.
CAD	computer-aided design.
CAE	computer-aided engineering.
CAO	conception assistée par ordinateur.
CATIA®	conception assistée tridimensionnelle interactive appliquée.
CFD	computational fluid dynamics.
CNAM	conservatoire national des arts et métiers.
D2S	dynamique des structures et des systèmes.
DES	detached eddy simulation.
DICE	deep inside computer experiments.
DNS	direct numerical simulation.
ECL	École Centrale de Lyon.
EDP	équation aux dérivées partielles.
ESI	engineering systems international.
FEA	finite element analysis.
FEM	finite element method.
FUI	fonds unique interministériel.
GIE	groupement d'intérêt économique.
HPC	high performance computing.
HVAC	heating, ventilation and air-conditioning.
ICFD	international conference on flow dynamics.

Liste des acronymes et abréviations

IDIAP	Institut Dalle Molle d'intelligence artificielle perceptive.
IGES	initial graphics exchange specification.
IHM	interface homme-machine.
INSA	institut national des sciences appliquées.
LaMCoS	laboratoire de mécanique des contacts et des structures.
LES	large eddy simulation.
LHS	Latin hypercube sampling.
LLE	local linear embedding.
LMSSC	laboratoire de mécanique des structures et des systèmes couplés.
LTDS	laboratoire de tribologie et dynamique des systèmes.
MAPAO	mise au point assistée par ordinateur.
MARS	multivariate adaptative regression spline.
Matlab [®]	matrix laboratory.
MDS	multidimensional scaling.
MSME	modélisation et simulation multi-échelle.
NURBS	non-uniform rational basis splines.
NVH	noise, vibration, and harshness.
ONERA	office national d'études et de recherches aérospatiales.
OW2 consortium	objectweb consortium and orientware.
PCA	principal component analysis.
PDE	partial differential equation.
PLEX	plan d'expériences.
polyMARS	polychotomous regression based on MARS .
PSA	Peugeot société anonyme.
PTC	parametric technology corporation.
RANS	Reynolds-averaged Navier-Stokes.
ReDICE	suite du projet DICE .
RODIN [®]	robust structural optimization for design in industry.
S2A	souffleries aéroacoustiques automobiles.
Scilab [®]	scientific laboratory.
SGE	simulation des grandes échelles.
SND	simulation numérique directe.
Star-CCM+ [®]	star-computational continuum mechanics+.
STB	spécifications techniques de besoin.

STD	spécifications techniques détaillées.
STEP	standard for the exchange of product model data.
STG	spécifications techniques générales.
STL	stereolithography.
STR	spécifications techniques de réalisation.
SVD	singular value decomposition.
SVM	support vector machines.
UPMC	Université Pierre et Marie Curie (Paris 6).
VB	visual basic.
WCCM	world congress on computational mechanics.
XML	extensible markup language.

AUTORISATION DE SOUTENANCE

Vu les dispositions de l'arrêté du 7 août 2006,

Vu la demande du Directeur de Thèse

Monsieur L. JEZEQUEL

et les rapports de

Monsieur C. SOIZE
Professeur - Université Paris Est - Laboratoire de Modélisation et Simulation Multi Echelle
Equipe de Mécanique - 5 bd Descartes - 77454 MARNE-LA-VALLÉE cedex 2

Et de

Monsieur R. OHAYON
Professeur - LMSSC - Chaire de Mécanique - Case 353 - 2 rue Conté - 75003 PARIS

Monsieur FROMENT Pierre

est autorisé à soutenir une thèse pour l'obtention du grade de **DOCTEUR**

Ecole doctorale MECANIQUE , ENERGETIQUE, GENIE CIVIL ET ACOUSTIQUE

Fait à Ecully, le 18 avril 2014

P/Le directeur de l'E.C.L.
La directrice des Etudes

