Contribution à la résolution de problèmes inverses sous contraintes et application de méthodes de conception robuste pour le dimensionnement de pièces mécaniques de turboréacteurs en phase avant-projets.

Maëva Biret

▶ **To cite this version:**

Maëva Biret. Contribution à la résolution de problèmes inverses sous contraintes et application de méthodes de conception robuste pour le dimensionnement de pièces mécaniques de turboréacteurs en phase avant-projets.. Statistiques [math.ST]. Université Pierre et Marie Curie - Paris VI, 2016. Français. NNT : 2016PA066294 . tel-01454988

HAL Id: tel-01454988
https://theses.hal.science/tel-01454988

Submitted on 10 Jul 2017

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

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

THÈSE

Pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITE PIERRE ET MARIE CURIE

ÉCOLE DOCTORALE SCIENCES MATHEMATIQUES DE PARIS CENTRE (ED 386)

DOMAINE DE RECHERCHE : STATISTIQUE THEORIQUE ET APPLIQUEE

Présentée par

Maëva BIRET

Contribution à la résolution de problèmes inverses sous contraintes et application de méthodes de conception robuste pour le dimensionnement de pièces mécaniques de turboréacteurs en phase avant-projets.

Directeur de thèse : **Pr. Michel BRONIATOWSKI**

Devant la Commission d'Examen

JURY

M. Michel BRONIATOWSKI	Professeur Université Pierre et Marie Curie	Directeur
M. Mohamed ACHIBI	Ingénieur statisticien Safran Aircraft Engine	Co-encadrant
M. Nikolaos Limnios	Professeur Université de Technologie de Compiègne	Rapporteur
M. Bertrand Iooss	Ingénieur chercheur sénior EDF	Rapporteur
M. Paul Deheuvels	Professeur Université Pierre et Marie Curie	Examinateur
M. Jérôme Lacaille	Expert émérite Safran	Examinateur
M. Olivier Lopez	Professeur Université Pierre et Marie Curie	Examinateur
M. Simon Charbonnier	Ingénieur mécanique Safran Aircraft Engines	Examinateur

Laboratoire de **S**tatistique **T**héorique et **A**ppliquée
Université Pierre et Marie Curie
Tour 15-25
4 Place Jussieu
75252 Paris

Safran Aircraft Engine
Site de Villaroche
Rond Point René Ravaud
77550 Moissy Cramayel

Remerciements

Mes remerciements sont adressés à toutes les personnes qui m'ont aidée, de près ou de loin dans la réalisation de ce mémoire de thèse. En effet, même si le diplôme est personnel, le travail de thèse est une aventure qui ne se vit pas en solitaire.

En premier lieu, je remercie M. BRONIATOWSKI, professeur à l'Université Pierre et Marie Curie. En tant que directeur de thèse, il a été présent à tous moments, des plus faciles aux plus difficiles. Son soutien et sa confiance furent souvent rassurants et motivants. Cette thèse n'aurait certainement pas été ce qu'elle est sans sa générosité et l'intérêt qu'il a porté à mon travail. De manière plus générale, je remercie l'ensemble du laboratoire (LSTA) et notamment M. BIAU, directeur du laboratoire, pour leur accueil.

Rapporteurs de ma thèse, M. IOOSS et M. LIMNIOS ont pris le temps de lire ce mémoire avec grande attention. Je les remercie pour leurs rapports bienveillants et constructifs. Merci également aux autres membres du jury, M. DEHEUVELS, M. LACAILLE, M. LOPEZ et M. CHARBONNIER, d'avoir accepté d'assister à ma soutenance.

Ces trois années, je les ai partagées avec toute l'équipe des *Méthodes et Outils* de la division *Intégration*, à Safran Aircraft Engines. Merci d'abord à M. GIRARD et M. BELEY de m'avoir accueillie dans leur équipe et d'avoir cru en mon projet. Un merci tout particulier à mon responsable de thèse, M. ACHIBI, sans qui cette thèse n'aurait même pas vu le jour, et aussi pour sa sympathie et ses nombreux conseils. Merci aussi à M. CHARBONNIER et M. PERRIER pour leur aide avec le cas test et à tous les membres de l'équipe pour leur bonne humeur, les concours de gâteaux et les pauses café : Yohan, Claire, Guillaume, Nedjma, Céline, Émilie, Thomas, Alix, Nicolas, Pierre-Yves, Marie-Océane, Alejandra.

Il ne faudrait pas que j'oublie les personnes qui sont tout de même les plus importantes dans

ma vie : ma famille. Un grand merci à mes parents pour m'avoir soutenue sans relâche durant toutes ces années. Merci également à ma grand-mère, sa joie de vivre me remonte toujours le moral. Et bien sûr, merci à Damien, l'amour de ma vie. Il m'a donné la force d'avancer et de mener à bien cette thèse. Sa confiance sans faille, sa patience lors des longues journées de rédaction, son soutien dans les moments difficiles et son amour m'ont permis d'accomplir un des plus grands projets de ma vie.

Un grand MERCI à tous !!

Table des matières

Table des figures

Liste des tableaux

Introduction générale

Le travail présenté dans ce mémoire est l'objet d'une collaboration entre le Laboratoire de Statistique Théorique et Appliquée (LSTA) de l'Université Pierre et Marie Curie (Paris 6) et l'entreprise Safran Aircraft Engine.

Safran Aircraft Engine, société du groupe SAFRAN, conçoit, développe, produit, et commercialise, seul ou en coopération, des moteurs pour avions civils et militaires, pour lanceurs spatiaux et pour satellites. Safran Aircraft Engine propose également aux compagnies aériennes, aux forces armées et aux opérateurs d'avions une gamme complète de services pour leurs moteurs aéronautiques. Face à la concurrence et à l'amélioration permanente des technologies, Safran Aircraft Engine doit rester un motoriste compétitif en proposant des nouveaux moteurs innovants qui sont moins bruyants, plus économes en consommation de carburant, plus légers, etc. Le développement d'un nouveau moteur se fait le plus souvent à la demande d'un avionneur qui projette de concevoir un nouvel appareil ou de remotoriser des appareils existants. La première phase du développement consiste à répondre à l'appel d'offre de l'avionneur, il s'agit de la phase avant-projets. Au terme de cette phase, les choix de conception, déterminants pour la suite du programme, sont validés et l'architecture du moteur est fixée. Phase clé de la conception d'un nouveau moteur, la phase avant-projets est un processus itératif faisant appel à plusieurs disciplines physiques comme la thermodynamique, l'aérodynamique et la mécanique. Chaque discipline veut optimiser ses performances tout en assurant la satisfaction des exigences de l'avionneur et de la réglementation. Afin de coordonner ces optimisations, les différents métiers doivent parfois échanger ensemble de nombreuses fois, c'est ce que l'on appelle des itérations (ou rebouclages). Ces itérations représentent des temps de calculs et des ressources humaines

importants. De plus, la phase avant-projets est également marquée par des variations importantes sur les entrées du système étudié, ce qui engendre de nombreuses incertitudes.
Dans le but de répondre aux appels d'offre des avionneurs avec des intervalles de confiance précis, les ingénieurs avant-projets ont un objectif permanent d'amélioration de la qualité de leurs résultats tout en réduisant la durée et les coûts de leurs études.

C'est dans ce contexte que cette thèse a été mise en place. L'objectif y est double. D'une part, il s'agit de définir une méthodologie de conception robuste dans le but de fournir des optima (masse, taille, coûts, consommation, etc) insensibles à des variations possibles (géométrie, caractéristiques matériaux, etc) lors du passage du dimensionnement à la conception. D'autre part, il s'agit de développer des méthodes de résolution de problèmes d'inversion mal posés afin de faciliter voire d'éviter les itérations en avant-projets. L'utilisation de ces méthodes d'optimisation et d'inversion nécessite une étude préliminaire de réduction de la dimension et de méta-modélisation permettant de réduire la complexité du problème et les temps de calculs pour la suite.

Ce mémoire est ainsi divisé en cinq chapitres. Dans le premier chapitre, les différentes phases de conception d'un nouveau moteur sont présentées brièvement et plus particulièrement la phase avant-projets. Le principe de fonctionnement d'un turboréacteur et plus précisément des pièces que nous avons étudiées dans nos cas tests y est précisé. Le contexte mathématique et statistique de l'étude y est également exposé. Dans ce même chapitre, les objectifs de la thèse sont détaillés ainsi que les deux cas tests utilisés. Le cas test principal porte sur le dimensionnement aéro-mécanique d'un compresseur HP en avant-projets. Le cas test secondaire consiste à établir une fonction de transfert pour ensuite retrouver l'effort appliqué sur un support-palier à partir des déformations mesurées lors d'un essai. Pour finir, la problématique est explicitée.
Le deuxième chapitre est consacré aux méthodes de réduction de la dimension et de méta-modélisation dont l'état de l'art est établi avec une partie en annexe. Quelques rappels sur les plans d'expériences, utiles pour l'application de ces méthodes, y sont également effectués. Les méthodes de réduction de la dimension et de méta-modélisation sont appliquées sur le cas test principal pour deux sorties d'intérêt. Le cas test secondaire permet de mettre en évidence les limites des méthodes usuelles de méta-modélisation.
Le troisième chapitre consiste à définir une méthodologie de conception robuste avec le recensement, la modélisation et la propagation des incertitudes, la formulation du problème de conception robuste et les méthodes d'optimisation multi-objectifs. Appliquée au cas test principal, cette méthodologie permet d'obtenir un optimum robuste de la masse du compresseur

HP.
Le quatrième chapitre regroupe les développements originaux de la thèse sur les méthodes de résolution de problèmes inverses. Après un exposé que nous espérons exhaustif de l'état de l'art des méthodes d'inversion, deux nouvelles méthodes sont décrites. L'intérêt est porté plus particulièrement sur une des deux méthodes, appelée méthode COMET et qui ne nécessite aucune hypothèse forte sur la fonction étudiée. Cette méthode a été testée jusqu'en dimension 4 et sur des fonctions variées, plus ou moins régulières. Appliquée sur le cas test principal, cette méthode permet de résoudre un problème d'intégration (collision entre deux pièces) directement sur le code de calculs. Elle fournit ainsi plusieurs solutions au problème inverse, solutions qui satisfont en plus toutes les contraintes. Le cas test secondaire met en évidence les limites des méthodes proposées. Il s'agit de résoudre un problème inverse ayant une unique solution. Pour un tel problème, les méthodes développées en thèse permettent de trouver la solution mais les méthodes usuelles de type Newton restent plus efficaces.
Le cinquième et dernier chapitre regroupe les bases théoriques d'une méthode probabiliste qui pourrait, à terme, être utilisée en optimisation ou pour résoudre des problèmes inverses. Rédigé en anglais, ce chapitre regroupe deux articles dont un a été publié par Springer dans un recueil en l'honneur de Paul Deheuvels. Le but de la méthode étudiée est d'établir une loi dont les observations seront toutes situées au-dessus d'une valeur choisie, voire très proches de cette valeur.

Contributions de la thèse

Le but de ce chapitre est de préciser les contributions originales de cette thèse, afin de guider le lecteur plus intéressé par les aspects théoriques que pratiques.

Le modèle présenté dans le Chapitre II pour le cas test secondaire est une proposition originale. A travers cet exemple apparaissent les limites des modèles usuels qui ne permettent pas toujours de prendre en compte les propriétés physiques du phénomène étudié. Un modèle paramétrique à deux variables explicatives a été construit. L'inconvénient est que la relation entre la variable à expliquer et une des variables explicatives doit avoir une forme particulière et être périodique. Cette particularité fait qu'il est difficile de prendre en compte la seconde variable explicative dans ce modèle. C'est pourquoi le modèle paramétrique établi possède des paramètres qui varient en fonction de cette seconde variable explicative. Ceci permet ainsi de construire un modèle dépendant des deux variables.

Les deux méthodes proposées dans le Chapitre IV sont des méthodes totalement nouvelles pour résoudre des problèmes inverses mal posés. Elles permettent d'obtenir plusieurs solutions à une équation qui peut par exemple en avoir une infinité. La première méthode proposée est la méthode MRM. Elle a fait l'objet d'un article accepté et présenté au Congrès Lambda Mu 19 à Dijon en 2014. Il est également référencé dans les actes du congrès dans la base CNRS I-revues (Biret et al., 2014). Cette méthode très efficace nécessite une hypothèse de monotonie de la fonction. La convergence de la méthode vers l'ensemble des solutions de l'équation à résoudre est disponible en Annexe C.
Pour éviter cette hypothèse forte de monotonie, une nouvelle méthode en deux variantes a été

développée : SAFIP et COMET. Cette méthode simple est applicable en toutes dimensions et sur de nombreux cas. Testée sur différentes fonctions plus ou moins régulières, cette méthode propose de nombreuses possibilités. La méthode SAFIP est une version basée sur des chaînes décroissantes vers la solution et fait l'objet d'un article (Biret and Broniatowski, 2016). La convergence de cette première version a été démontrée. Un article sur la méthode COMET a été accepté pour une présentation au Lambda-Mu 20 qui a eu lieu à Saint-Malo en octobre 2016. Il sera référencé dans la base CNRS I-revues. Elle fait également l'objet d'un brevet Safran Aircraft Engine portant sur la résolution du cas test industriel.

Le Chapitre V s'inscrit dans la suite des travaux de (Broniatowski et al., 2014) sur l'étude d'une marche aléatoire conditionnée par des grandes déviations. Le chapitre est un regroupement de deux articles. Le premier consiste à étudier les propriétés de la loi tiltée qui, pour des queues légères et sous des conditions de régularité, approxime la loi de la marche aléatoire conditionnée. Cet article, (Biret et al., 2015), a été publié par Springer dans un recueil d'articles en l'honneur du Professeur Paul Deheuvels. Le second article, (Biret et al., 2016), porte sur l'étude d'un théorème de Gibbs conditionnel pour la loi de la marche aléatoire conditionnelle . Sous les mêmes conditions de régularité que l'article précédent, plusieurs résultats sont proposés.

Chapitre I

Problématique industrielle et objectifs de la thèse

1 Contexte industriel

L'aéronautique est un secteur très concurrentiel où les motoristes doivent répondre aux exigences des avionneurs et respecter une réglementation stricte. Les premiers disposent de l'offre : ils conçoivent, développent, produisent et commercialisent les moteurs d'avions. Les seconds disposent de la demande : ils développent et produisent les avions pour lesquels un certain type de moteur est exigé.

1.1 Développement d'un moteur

Le développement d'un nouveau moteur découle le plus souvent d'une demande d'un avionneur dans le but de développer un nouvel appareil ou de remotoriser des appareils existants.

1.1.1 Développement en quatre phases

La définition des spécifications du moteur par l'avionneur constitue les prémisses du développement d'un nouveau moteur que l'on peut décomposer en quatre grandes phases :

- *phase amont de la conception :* phase assurant la réponse à l'appel d'offre de l'avionneur (RFP pour *Request For Proposal*), la réalisation des spécifications des performances du moteur et fournissant une première idée de l'architecture, des coûts, etc,

- *phase de définition :* établir l'architecture et la validation des concepts et des solutions technologiques du moteur,
- *phase de conception, d'industrialisation et de validation :* l'architecture et les spécifications techniques du moteur sont figées, les liasses du moteur sont créées, la fabrication des pièces commence et la validation est effectuée avec des essais au sol et en vol,
- *phase de mise en service :* livraison du premier moteur et validation en situation réelle.

Les activités de chaque phase du développement sont détaillées dans la Figure I.1 issue de la thèse de (NguyenVan, 2006). Elles sont regroupées dans les six rectangles successifs. Les phases sont jalonnées d'étapes de validation pour assurer que les objectifs de chaque activité soient atteints. Il s'agit, sur la Figure I.1, des triangles situés sous les activités. Le tout premier triangle représente l'étape consistant à établir l'équipe de direction de programme (EDP). Les suivants sont situés à la fin de chaque activité. Les actions représentées par des losanges sont celles en lien avec les avionneurs. Nous avons parlé de la RFP lors de la phase amont. Nous trouvons aussi, par exemple, le premier moteur à être testé (FETT pour *First Engine To Test*) dans la phase de conception ou encore la livraison du premier moteur à l'avionneur à la fin du processus de développement.

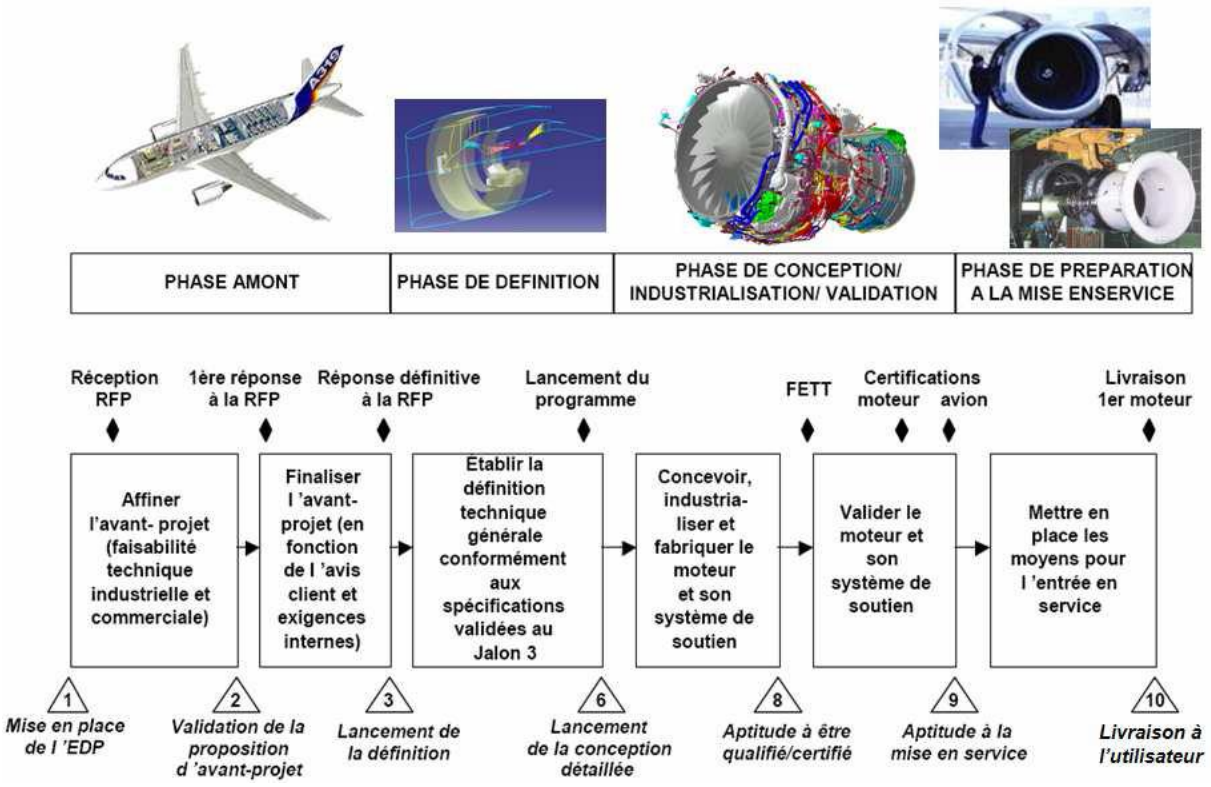

Figure I.1 – Description des phases de développement d'un moteur et les jalons de validation (NguyenVan, 2006)

Le travail mené en thèse a porté sur un cas d'application en phase amont, phase dont nous détaillons le principe dans la section suivante.

1.1.2 Phase amont de la conception : la phase avant-projets

Nous nous intéressons à la première phase de développement : la phase amont de la conception ou phase avant-projets. Cette phase se positionne comme une phase clé dans la conception de moteurs d'avions. En effet, les grandes orientations du programme moteur y sont définies. À partir des spécifications fournies par l'avionneur (poussée, consommation, émissions de CO_2, masse, bruit...), des choix technologiques sont effectués (réponse définitive à la RFP) et des engagements contractuels sont pris (lancement du programme). Au terme de cette phase, l'architecture du moteur est figée et la conception détaillée des différents composants peut commencer.

La phase avant-projets est un processus itératif qui peut être divisée en trois disciplines majeures, chacune propre à un domaine spécifique : la thermodynamique, l'aérodynamique et la mécanique. La Figure I.2 représente ces trois étapes, vues chacune comme un système (on parlera aussi de boîte) composé de variables d'entrée, d'un code de calcul et de variables de sortie. En entrée de la boîte thermodynamique, on retrouve les spécifications des avionneurs ainsi que

Figure I.2 – Trois étapes majeures de la phase avant-projets

des critères préliminaires fixés par les ingénieurs vis à vis du marché, de la concurrence et des attentes des futurs clients. Cette première étape permet de déterminer ce que l'on appelle le cycle thermodynamique. Ce cycle est un jeu de variables représentant les performances (températures, pressions, débits, régime) à différents points de vol du moteur (cf. Figure I.7). Ces variables constituent les sorties de cette première boîte et donc les entrées de la deuxième, la boîte aérodynamique. Durant cette deuxième étape, la veine aérodynamique du moteur est générée. Il s'agit d'un jeu de données représentant la géométrie du canal dans lequel le flux d'air circule à travers les principaux composants du moteur. Ce canal peut être représenté schématiquement à la Figure I.3.

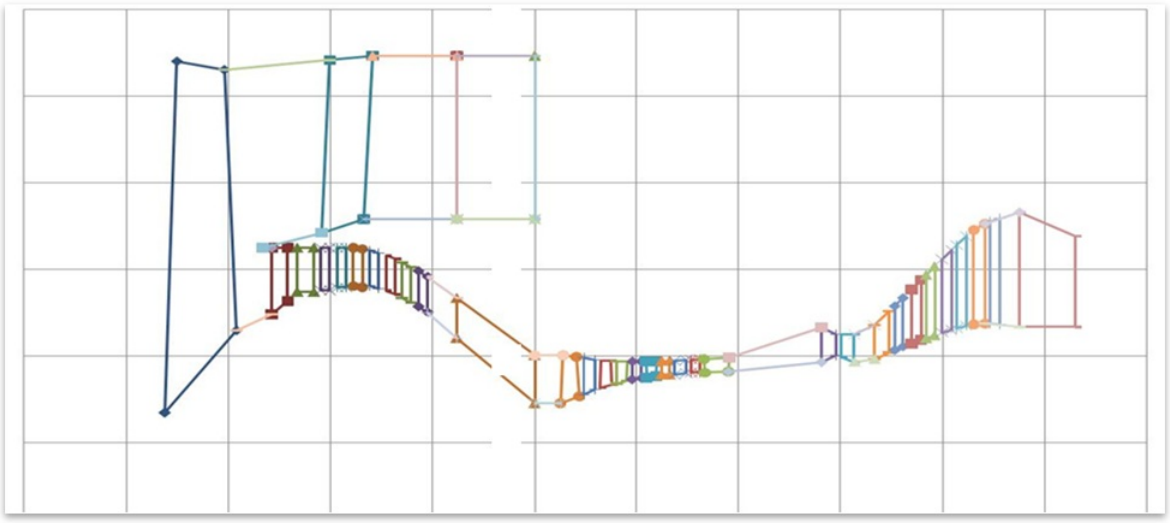

Figure I.3 – Illustration de la veine complète d'un moteur

Ces données s'obtiennent en sortie de la boîte aérodynamique et sont utilisées en entrée de la boîte mécanique. Durant cette dernière étape, l'architecture du moteur est définie et les différents composants sont dimensionnés. On obtient alors la masse globale du moteur par optimisation mécanique.

Cette phase avant-projets se caractérise par trois particularités :

- des variations importantes des données d'entrée peuvent survenir lors du dimensionnement. Les causes de ces variations peuvent être internes (stratégie, hypothèses simplificatrices) ou externes (évolution des spécifications avionneur). Lors de la recherche d'optimum, ces variations potentielles doivent être prises en compte pour garantir la robustesse de l'optimum retenu par rapport aux variations (bornées) des paramètres d'entrée ;
- afin de conserver une bonne réactivité, les modèles utilisés sont basés sur des hypothèses simplificatrices introduisant un biais non négligeable par rapport aux résultats pouvant être mesurés en essais. Cette incertitude sur les résultats de sortie doit elle-même être évaluée, de manière à guider les choix vers des solutions qui ne seront pas remises en cause lors de l'avancement du programme ;
- les trois étapes majeures de la phase avant-projets (cf Figure I.2) ne s'enchaînent pas de façon linéaire, certains « rebouclages » entre les différents métiers sont nécessaires pour assurer les spécifications. Ces rebouclages sont illustrés à la Figure I.4. Par exemple, l'exécution linéaire des trois étapes du dimensionnement peut conduire à un problème d'intégration. Il s'agit d'une infaisabilité mécanique où deux pièces sont en collision, on

parle alors de clash. Si le problème ne peut pas être résolu au niveau mécanique, il faut alors remonter au niveau aérodynamique en modifiant la veine, voire au niveau thermodynamique en modifiant le cycle. Dans le premier cas, seules les deux dernières étapes sont relancées. Dans le second cas, il faut refaire les trois étapes. Notons qu'il n'est pas évident a priori de pouvoir résoudre le problème d'un point de vue aérodynamique ou thermodynamique, d'où un nombre parfois important de rebouclages.

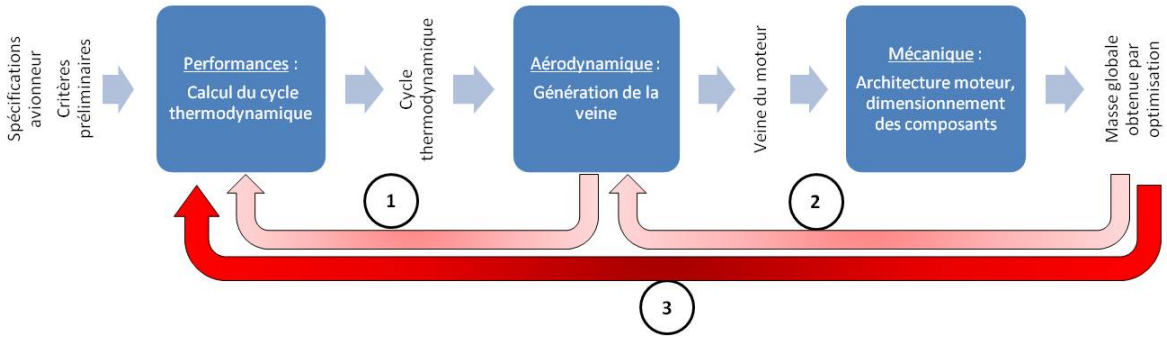

Figure I.4 – Rebouclages entre les trois boîtes majeures de dimensionnement de la phase avant-projets

Face à cela, deux problèmes majeurs se posent lors de la phase avant-projets :

- peu de données sont figées à ce stade du programme et de nombreuses solutions peuvent être envisagées. Les modèles de dimensionnement en avant-projets se caractérisent donc par un grand nombre de variables d'entrée et de nombreuses évaluations du code de calcul ;
- de nombreuses contraintes géométriques et métier sur les sorties du système sont à respecter. Certaines peuvent être prises en compte dès le début, à travers les variables d'entrée par la construction même de l'outil. Il s'agit par exemple des contraintes qui dépendent des entrées selon une équation connue. Dans l'outil, le calcul n'est effectué que si cette équation est satisfaite. Pour d'autres, leur satisfaction ne sera connue qu'après l'évaluation du code de calculs. C'est le cas des contraintes dont l'expression n'est pas connue. Elles dépendent donc des entrées mais d'une manière trop complexe pour l'exprimer directement dans l'outil. Les contraintes géométriques d'admissibilité présentes sur les disques sont dans ce cas.

Dans le but de répondre aux appels d'offre des avionneurs avec des incertitudes estimées et bornées, les ingénieurs avant-projets ont un objectif permanent d'amélioration de la qualité de leurs résultats tout en réduisant la durée et les coûts de leurs études.

Cette thèse s'inscrit dans cet objectif par deux aspects : la conception robuste et l'inversion de

fonction. En effet, la conception robuste consiste à optimiser une quantité d'intérêt (consommation, masse, coûts, etc) de façon à ce que l'optimum soit insensible à des variations possibles en conception. Il faut donc prendre en compte les incertitudes rencontrées durant tout le processus avant-projets. L'inversion de fonction consiste à résoudre par exemple les problèmes d'intégration mécanique afin de faciliter voire de supprimer les « rebouclages » entre les différents métiers.
Ceci constitue les deux principales problématiques de cette thèse. Mais avant de rentrer dans le vif du sujet, nous allons décrire assez brièvement le principe de fonctionnement des composants étudiés dans les cas tests pour un turboréacteur, moteur utilisé pour l'aviation civile. Puis nous présenterons les objectifs de la thèse et le cas test qui a servi à réaliser ce travail.

1.2 Principe de fonctionnement d'un turboréacteur

Utilisé pour la propulsion des avions à réaction civils et militaires, le turboréacteur fonctionne selon le principe d'action-réaction, illustré sur la Figure I.5 par l'exemple du ballon de baudruche. Lorsque le ballon est gonflé, la résultante des forces de pression est nulle, le ballon est en équilibre. Si un orifice est pratiqué, l'air s'échappe avec une certaine vitesse (action). Le ballon se déplace en sens inverse (réaction). La force est fonction de la masse évacuée et de la vitesse. L'objectif est d'accélérer une masse d'air en l'éjectant vers l'arrière afin de créer une

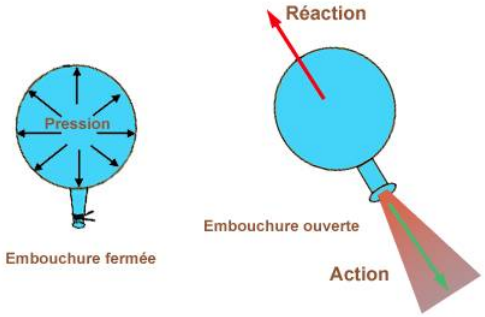

Figure I.5 – Principe d'action-réaction

augmentation de la quantité de mouvement. Il en résulte un déplacement du corps dans le sens opposé à l'éjection des gaz. C'est le principe des turboréacteurs pour lesquels un jet propulsif est créé par compression puis par chauffage de l'air. Le fonctionnement des turboréacteurs se fait selon le schéma de la Figure I.6. Le comburant, l'air, subit des séries de transformations suivant trois phases principales : une compression, puis une combustion à l'aide d'un carburant et enfin une détente afin de créer une énergie (cinétique donnée par la vitesse, mécanique donnée par la position) qui va permettre la propulsion. On peut également illustrer ce fonctionnement par ce

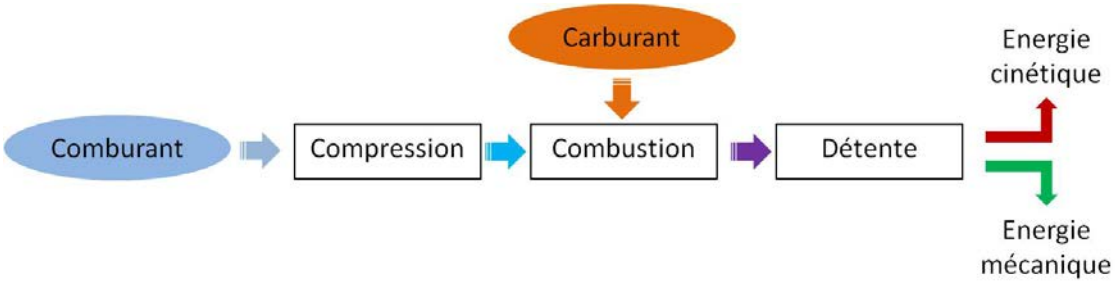

Figure I.6 – Principe de fonctionnement des turboréacteurs

qu'il est convenu d'appeler le « diagramme d'écoulement des gaz » (cf Figure I.7) qui montre l'évolution de trois paramètres : la pression P en bleu, la vitesse V en vert et la température t ° en rouge.

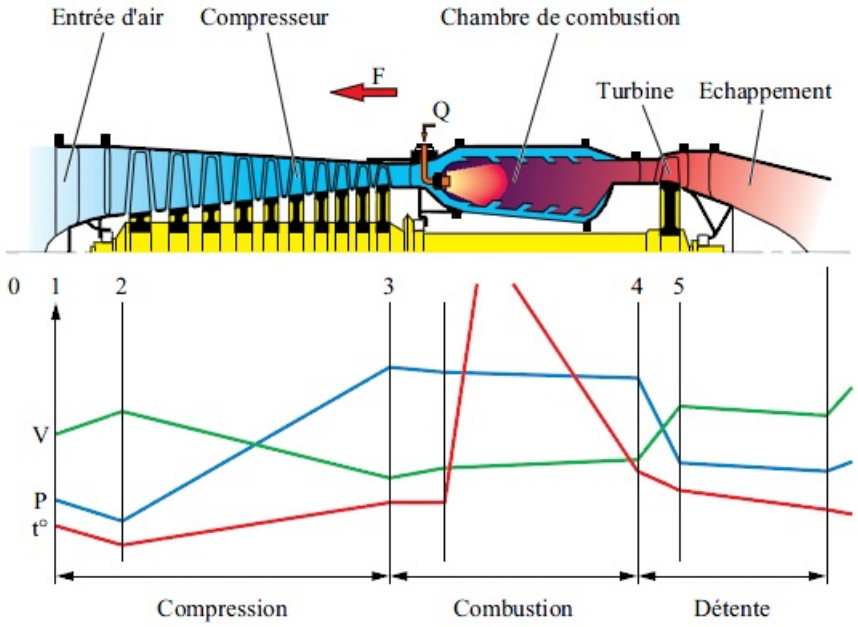

Figure I.7 – Diagramme d'écoulement des gaz

Après le passage de l'air dans les différentes parties du turboréacteur, les gaz sont évacués avec une grande vitesse, ce qui produit la poussée F.
Il existe plusieurs types de turboréacteurs selon le nombre de flux et le nombre de corps.

1.2.1 Turboréacteur « mono-flux et simple-corps »

C'est le cas le plus simple. Un compresseur, couplé à une turbine, assure la compression de l'air capté via une manche d'entrée. La turbine est elle-même entraînée par les gaz chauds

qui sortent d'une chambre de combustion. Dans cette chambre, l'énergie est fournie par la combustion de kérosène avec l'oxygène disponible dans l'air absorbé (et comprimé). Une fois que l'énergie mécanique nécessaire à l'entraînement du compresseur a été prélevée, il reste encore suffisamment d'énergie provenant de la combustion pour fournir l'énergie de propulsion requise. La poussée est finalement obtenue au travers d'une tuyère destinée à accélérer les gaz sortant de la turbine et dont la section d'éjection est réglée de telle sorte que la poussée soit optimale. Cette section peut être définie une fois pour toutes ou ajustable en vol.

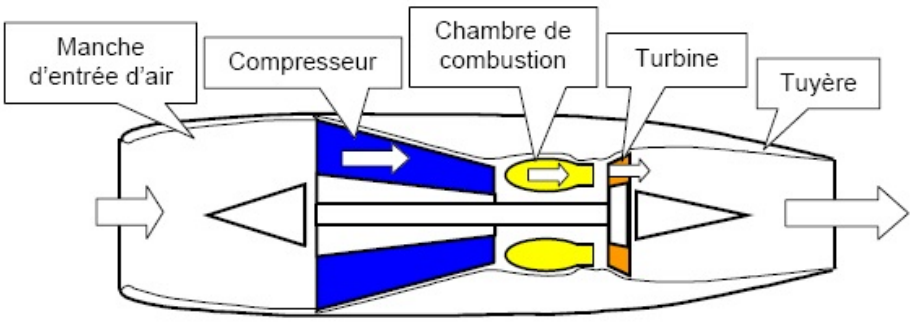

Figure I.8 – Principaux constituants d'un turboréacteur mono-flux et simple-corps (Thévenin, 2004)

1.2.2 Turboréacteur « double-flux, simple-corps »

Dans un turboréacteur à simple flux, les gaz éjectés le sont en général à des vitesses très élevées. La poussée est également très élevée mais, en contrepartie, on perd une grande partie de l'énergie cinétique du jet dont les particules d'air et de gaz brûlés qui sortent du moteur à grandes vitesses vont se disperser dans l'air ambiant. Pour réduire ces pertes d'éjection, la solution consiste à prélever une partie de l'air en amont du compresseur haute pression (flux secondaire), puis de le mélanger au flux sortant de la turbine avant son éjection proprement dite à l'aval du moteur vers l'extérieur. Ceci créera donc une poussée supplémentaire en accélérant le flux d'air secondaire (qui ne passera pas par la chambre de combustion) à une vitesse modérée, mais avec un débit significatif. La Figure I.9 en illustre ce principe.

1.2.3 Turboréacteur « double-flux , à double, voire triple-corps »

Le raisonnement est semblable à celui du cas précédent mais il est poussé beaucoup plus loin. Dans ce type de moteur, on cherche à obtenir du flux secondaire un maximum de poussée avec des vitesses d'éjection pas trop élevées. Le compresseur basse pression du cas précédent devient une soufflante de grand diamètre, très supérieur à celui du compresseur de base. Si cette

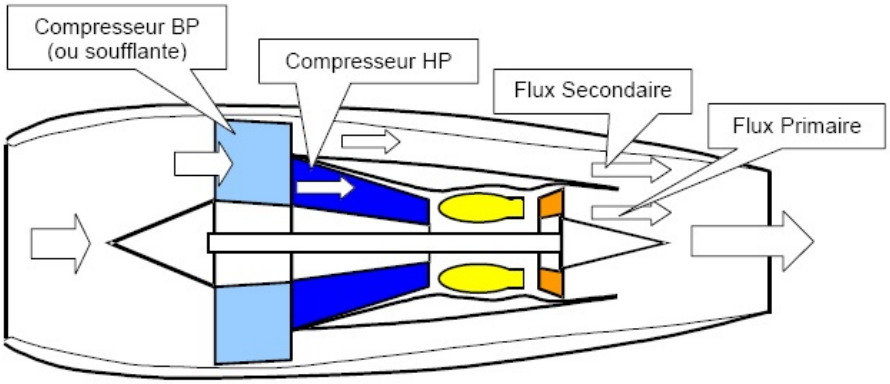

Figure I.9 – Turboréacteur double-flux, simple-corps (Thévenin, 2004)

soufflante devait être entraînée par la turbine de base, au même régime de rotation, les vitesses aérodynamiques que l'on rencontrerait au rayon extérieur de la soufflante seraient trop élevées pour obtenir un fonctionnement efficace. La solution consiste donc à entraîner cette soufflante par une turbine tournant plus lentement. Ceci est illustré sur la Figure I.10, dans le cas d'un double-corps.

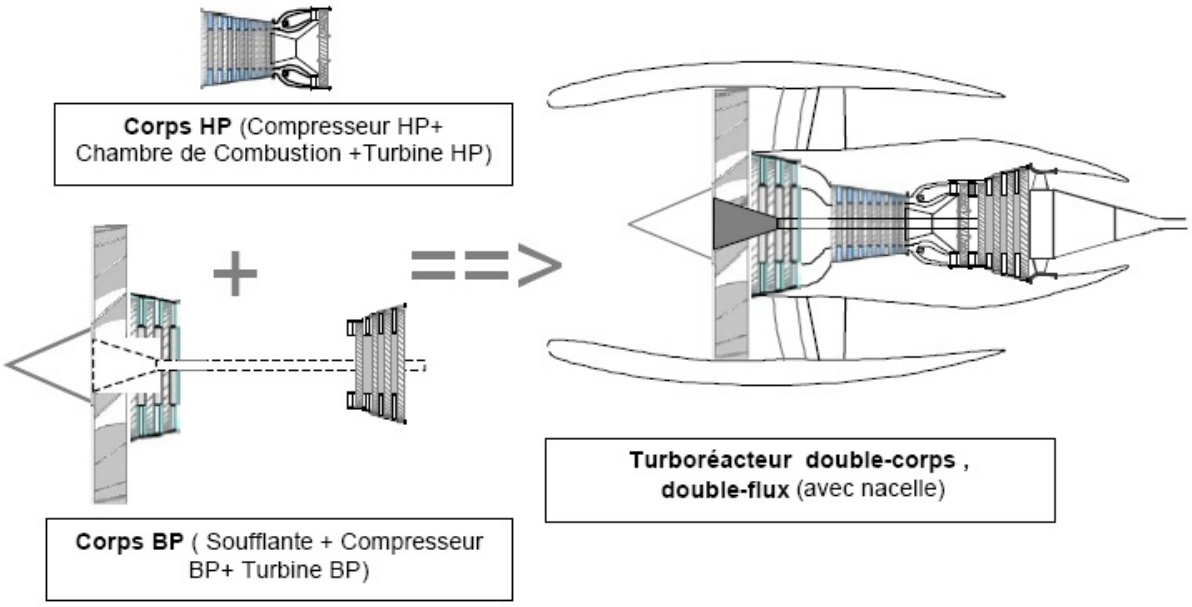

Figure I.10 – Principe de constitution d'un turboréacteur double-flux, double-corps (Thévenin, 2004)

Les turboréacteurs qui propulsent les avions de transports civils subsoniques modernes sont souvent du type double-corps, double-flux. En effet, l'avantage du double flux est que la consommation du moteur est beaucoup plus basse. La soufflante reçoit la totalité de l'air qui pénètre dans le moteur. Une grande partie de cet air va constituer le flux secondaire et générer la grande

majorité de la poussée, l'autre partie va constituer le flux primaire. Ce dernier passe par un compresseur basse pression (BP) solidaire de la soufflante, par un compresseur haute pression (HP), la chambre de combustion, puis la turbine HP et, pour finir, par la turbine BP avant d'être éjecté. Le flux secondaire est éjecté par une tuyère secondaire, le flux primaire par une tuyère primaire, sauf dans le cas où les deux flux sont mélangés. Dans ce dernier cas, les deux flux sont éjectés par une tuyère commune. Le dessin de la Figure I.11 donne une image d'un turboréacteur de ce type (cas des flux séparés).

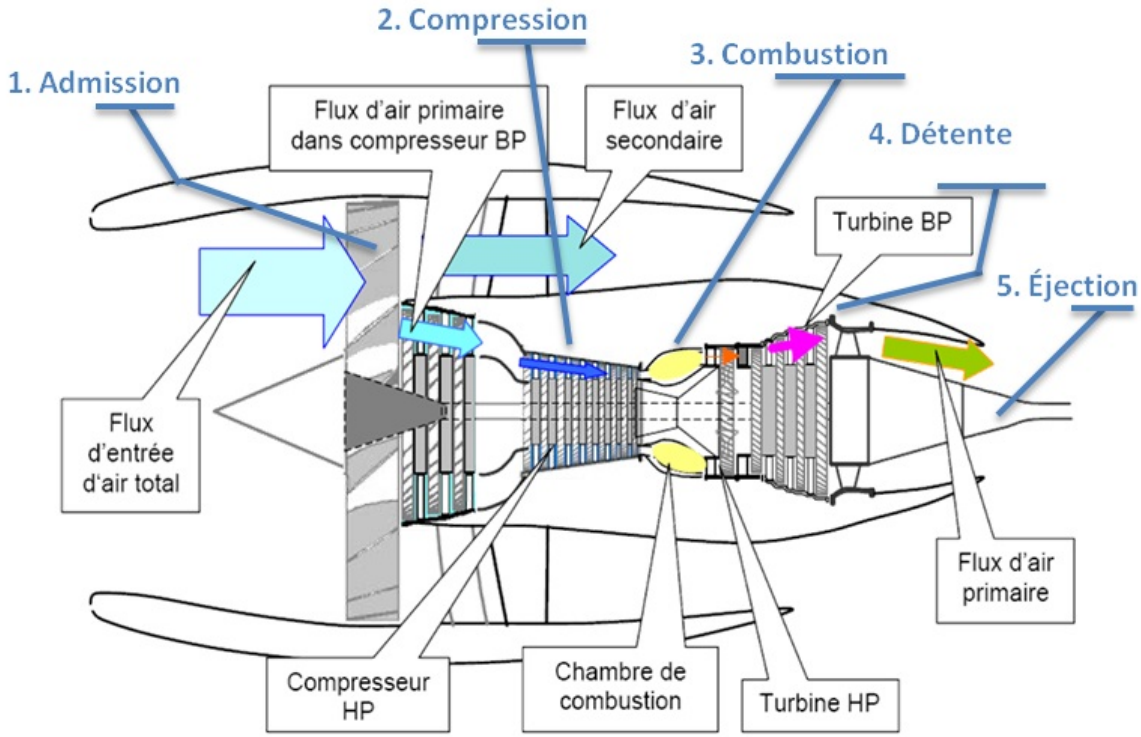

Figure I.11 – Turboréacteur double-flux, double-corps du type le plus courant (Thévenin, 2004)

1.2.4 Principe de fonctionnement d'un compresseur axial

Il existe trois types de compresseurs :

- les compresseurs axiaux et les soufflantes,
- les compresseurs centrifuges,
- les compresseurs axialo-centrifuges.

Le compresseur HP d'un turboréacteur est souvent un compresseur axial. Ce type de compresseur résulte d'un empilage d'« étages » composés chacun d'un « aubage mobile » et d'un « aubage fixe ».

Considérons d'abord un tel étage de compresseur. L'aubage mobile (ou roue) est constitué d'un disque circulaire sur lequel sont fixées des « aubes » qui ressemblent à des petites ailes (ailettes). Il tourne devant l'aubage fixe, circulaire (ou grille fixe), qui est également constitué d'aubes, fixes celles-ci.
La compression de l'air s'y passe en deux phases :

1. l'aubage mobile procure une accélération aux particules d'air en les déviant par rapport à l'axe moteur ;
2. l'aubage fixe qui le suit ralentit ces particules et transforme une partie de leur vitesse en pression. Cet aubage s'appelle aussi « redresseur » car il ramène l'écoulement de l'air, accéléré par l'aubage mobile, dans l'axe du moteur.

La Figure I.12 illustre ce fonctionnement. Sur le plan aérodynamique, les performances d'un

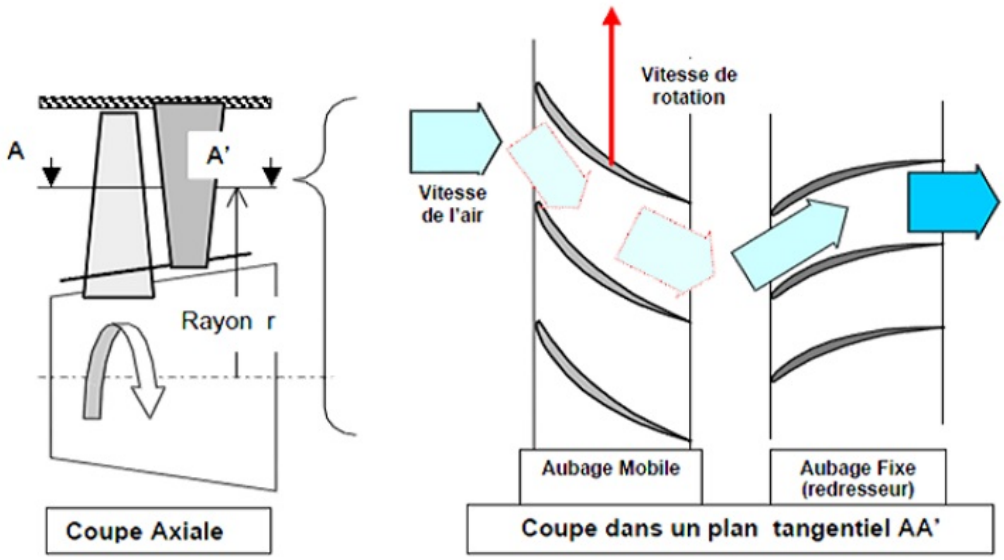

Figure I.12 – Schéma du fonctionnement d'un étage de compresseur

étage de compresseur sont caractérisées par trois grandeurs : son débit d'air, son taux de compression (ou rapport de pression) et son rendement, le tout étant assuré par la vitesse de rotation.

Un compresseur complet possède toute une succession d'étages, dont l'allongement des aubes (c'est-à-dire leur hauteur rapportée au diamètre de la roue ou de l'aubage) est de plus en plus faible au fur et à mesure qu'on progresse dans le compresseur (cf Figure I.13). L'ensemble des aubages mobiles constitue en partie ce que l'on appelle le rotor, l'ensemble des aubages fixes appartiennent au stator.

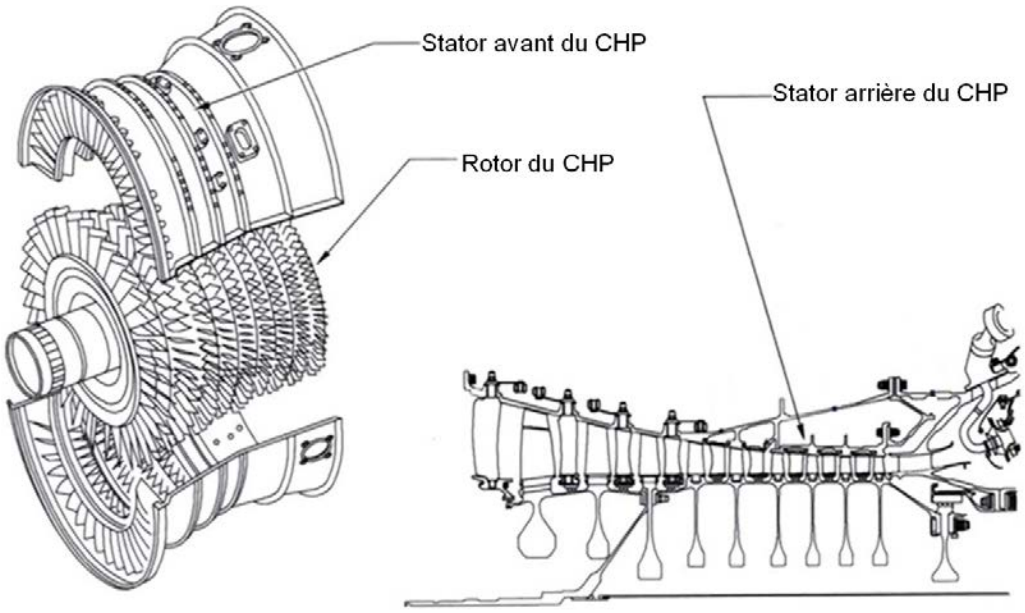

Figure I.13 – Vues en éclatée et schématique (coupe axialo-radiale) d'un compresseur HP

1.2.5 Rôle des paliers

Les arbres qui relient le compresseur et la turbine doivent pouvoir tourner à très grandes vitesses. Leur tenue mécanique est fondamentale. Même à grandes vitesses, ils doivent pouvoir tourner et rester rigoureusement rectilignes, parfaitement dans l'axe moteur. Ils doivent pouvoir supporter des efforts de plusieurs tonnes ainsi qu'un minimum de balourds (déséquilibres dynamiques d'un ensemble tournant) accidentels.
Des « paliers », constitués de roulements à billes ou à rouleaux, soutiennent ces arbres en s'appuyant sur les structures fixes du turboréacteur. Selon le nombre de corps, simple, double ou triple, les paliers sont respectivement au nombre de trois, quatre (ou cinq) et huit (ou neuf). Dans le cas des moteurs à corps multiples, il peut y avoir plusieurs paliers inter-arbres.
Pour éviter le déplacement vers l'avant ou vers l'arrière des ensembles mobiles, le palier amont (comme le palier 3 de la Figure I.14) est un palier de butée, monté sur roulement à billes.

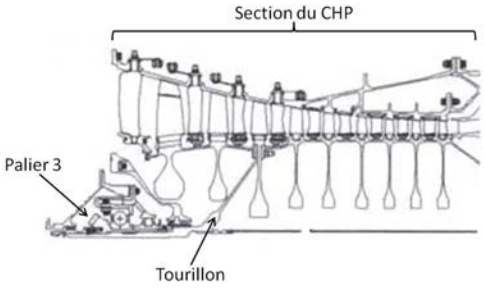

Figure I.14 – Coupe axialo-radiale d'un compresseur HP avec le palier 3 et le tourillon

Sous le palier 3 se trouve une pièce appelée tourillon. Elle assure la liaison mécanique entre le palier 3 et le rotor du CHP afin d'assurer sa rotation. Sur la Figure I.14, le tourillon est fixé sur le troisième disque du CHP.

L'étude présentée dans cette thèse porte sur l'optimisation robuste de la masse d'un compresseur HP et sur la résolution de problèmes d'intégration mécanique. Nous prenons comme exemple le problème d'intégration du palier 3 et du tourillon sous le compresseur HP. Les variables de clashs doivent être positives. Si l'outil fournit une valeur négative, alors il y a un problème d'intégration, ce qui signifie que deux pièces sont en collision.

1.3 Importance de la masse

Les motoristes, pour vendre de nouveaux moteurs aux avionneurs, visent sans cesse des objectifs d'amélioration sur différents aspects. Il peut s'agir des performances du moteur, de sa durée de vie, de sa consommation de carburant, des émissions de gaz polluants, de la réduction du bruit ou encore des coûts de production et de maintenance. Un de ces objectifs est souvent la masse du moteur, qui représente un des enjeux majeurs de la vente de moteurs aux avionneurs. En effet, la conception d'un avion dépend énormément de la masse du moteur. Par exemple, quand les moteurs sont sous les ailes, plus les moteurs sont lourds, plus les ailes doivent être renforcées donc agrandies, plus l'avion est lourd et plus il consommera de carburant. Une partie des négociations avec les avionneurs se fait sur la masse du moteur, dont l'estimation fournie à la fin de la phase avant-projets se doit d'être la plus précise possible. En effet, le motoriste pourra être amené à verser des indemnités si la masse réelle dépasse la masse agréée.

La masse du moteur nu est distinguée de la masse du moteur équipé, qui correspond à la masse du moteur nu plus celle des équipements. En général, la masse fournie est la masse PPS (*Powerplant Propulsive System*), qui correspond à la masse du moteur équipé à laquelle on

ajoute la masse de la nacelle (support et capots du moteur) et la masse EBU (*Engine Built Up*, c'est-à-dire l'ensemble des équipements montés sur le moteur par l'avionneur).

Pour le plus gros des turboréacteurs, la masse PPS est de l'ordre de 7500 kg, soit environ 50 fois plus élevée que celle du moteur d'une automobile moyenne. Rapporté à sa masse, un turboréacteur est à peu près 20 fois plus puissant qu'un moteur automobile.

2 Contexte mathématique et statistique

Le contexte industriel peut être traduit en des termes mathématiques et statistiques. Ceci est indispensable pour formuler le problème et y apporter des solutions mathématiques. Dans un premier temps, le code de calcul physique, qui regroupe les trois boîtes de dimensionnement illustrées à la Figure I.2, peut être représenté comme une boîte blanche.

2.1 Modèle boîte blanche d'un système

Une boîte blanche (ou boîte transparente) est un système dont on peut prévoir le fonctionnement interne car les caractéristiques de fonctionnement sont connues pour l'ensemble des éléments qui le composent. Notre système est composé d'outils dont le fonctionnement est bien connu puisqu'ils ont été développés à Safran Aircraft Engine. Nous pouvons donc avoir accès à tous les codes permettant d'obtenir les sorties du système. La Figure I.15 résume notre vision du système étudié et précise quelques notations.

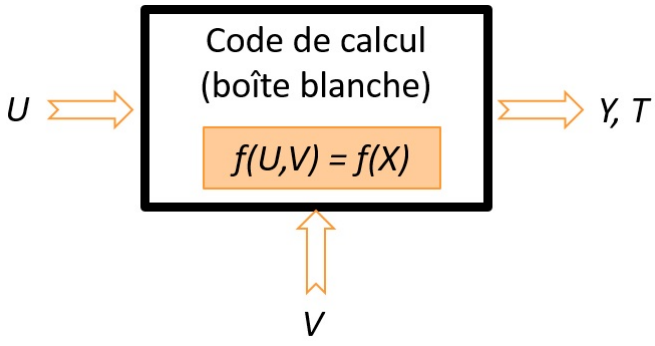

Figure I.15 – Vision boîte blanche

Le vecteur U regroupe les d_1 entrées du système. Ce sont des variables de décision, maîtrisables dans le système mais jamais totalement maîtrisées, car elles dépendent de l'environnement dans

lequel le système se trouve. Des variables géométriques sont un exemple de variables de décision. Le vecteur V regroupe les d_2 paramètres environnementaux du système. La température ou la pression de l'air sont des paramètres environnementaux. Afin d'alléger les notations, nous rassemblons ces deux types d'entrées dans le vecteur $X = (U, V)$ qui contient donc $d = d_1 + d_2$ variables. Le vecteur Y regroupe les k sorties du système tel que $Y = f(X)$. La fonction f est appelée fonction objectif. Bien que cette fonction soit déterministe, le vecteur Y est aléatoire à cause du caractère incertain de X et du biais apporté par l'utilisation de f vis-à-vis d'autres modèles possibles et vis-à-vis de la réalité. Dans notre situation, Y est une sortie vectorielle. Elle contient la masse du compresseur HP et sa géométrie. Les objectifs en avant-projets sont de minimiser la masse et de fournir une géométrie fonctionnelle des pièces. Ceci est assuré par le vecteur T regroupant les m contraintes qui doivent être satisfaites dans toutes les conditions d'utilisation. On notera $T = g(X)$ et on définira ainsi l'espace de conception par les m relations $T_i = g_i(X) \geq 0, i = 1, ..., m$. Les fonction g_i sont appelées fonctions contraintes.

2.2 Représentation du dimensionnement en avant-projets

Nous pouvons considérer le processus de dimensionnement thermo-aéro-mécanique du compresseur HP de différentes manières :

- le dimensionnement complet est représenté par une seule boîte. Les entrées sont alors les entrées du dimensionnement thermodynamique et les sorties sont celles du dimensionnement mécanique. La boîte contient tout le reste de la Figure I.2,
- chaque boîte de la Figure I.2 est vue comme une boîte individuelle. Chaque dimensionnement est ainsi représenté par une boîte dont les entrées/sorties sont celles des différents dimensionnements de la Figure I.2,
- les boîtes sont regroupées deux à deux (thermo-aérodynamique et aéro-mécanique).

Le choix de la représentation se fait suivant la problématique à résoudre. Jusqu'à présent, les dimensionnements étaient effectués séparément et représentaient donc trois boîtes successives. Mais le problème des rebouclages chronophages en ressources informatiques et humaines ne pourra être réglé que si nous considérons les boîtes deux à deux voire les trois ensemble. Les associations deux à deux permettront de traiter les rebouclages entre métiers successifs (mécanique vers aérodynamique ou aérodynamique vers thermodynamique), ce sont les flèches 1 et 2 sur la Figure I.4. Considérer une boîte pour les trois dimensionnements permettra de reboucler directement de la mécanique vers la thermodynamique, il s'agit de la flèche 3 sur la Figure I.4.

3 Objectifs de la thèse

L'objectif général de la thèse est double. D'un côté, il s'agit de développer, par la conception robuste, une démarche et des outils génériques probabilistes, fournissant une aide au choix d'une architecture robuste en avant-projets. D'un autre côté, il s'agit de développer des méthodes de résolution de problèmes inverses pour des fonctions à valeurs réelles et de plusieurs variables afin de résoudre rapidement des problèmes d'intégration en avant-projets. Ces deux objectifs font appel à des méthodes pouvant être de plus en plus complexes avec l'augmentation de la dimension du problème et donc coûteuses en temps de calculs. D'autant que le code de calculs lui-même peut être coûteux. Ceci nous mène au premier objectif de la thèse.

3.1 Réduire la dimension du problème et établir un modèle mathématique

Les codes de calcul complexes sont utilisés pour simuler, par exemple, un phénomène physique et sont considérés par l'utilisateur comme étant le modèle de référence du système étudié. Comme nous l'avons vu dans la section 2.1, ce système est vu comme une boîte blanche constituée d'un ensemble de variables d'entrée X, d'un ensemble de variables de sortie Y et du code de calcul f qui représente le modèle physique.
Ces codes de calculs sont utilisés pour traiter les problèmes usuels d'incertitudes, de sensibilité, d'optimisation ou encore de robustesse par exemple. Mais leur complexité peut rendre ces traitements difficiles, notamment à cause de temps de calculs importants. En effet, avec l'augmentation des puissances de calcul et l'amélioration des outils de simulation, la précision des modèles numériques est en pleine croissance, ce qui permet de se rapprocher de plus en plus des phénomènes physiques réels. Ceci nécessite un apport conséquent de connaissances, comme la prise en compte d'un très grand nombre de variables d'entrée. Ce nombre représente ce que l'on appelle la dimension du système, notée d.

Dans les problèmes à très grandes dimensions, certains phénomènes, inexistants en plus petite dimension, peuvent apparaître lors de l'analyse et le traitement des données. C'est ce que l'on appelle le fléau de la dimension (*curse of dimensionality* en anglais). L'idée est que le volume de l'espace d'entrée croît très rapidement avec la dimension. Ainsi, plus l'espace est grand, plus les données qui y sont représentées sont dispersées (*sparsity* en anglais). Pour tout traitement d'un système, les résultats ne peuvent être statistiquement fiables et solides qu'avec un nombre suffisant de données, nombre qui augmente de façon exponentielle avec la dimension

de l'espace d'étude.

Par exemple en échantillonnage, 10^2 points équi-répartis sont nécessaires pour couvrir l'intervalle unidimensionnel $[0, 1]$ avec une distance d'au plus 0.01 entre les points. Dans l'espace à 10 dimensions $[0, 1]^{10}$, ces 100 points sont alors des points isolés dans un espace apparaissant comme vide. Pour obtenir une couverture équivalente à celle des 100 points dans $[0, 1]$ (distance d'au plus 0.01 entre les points), il faut $10^{20} = (10^2)^{10}$ observations équi-réparties. Pour un espacement de 10^{-2}, l'hypercube de dimension 10 apparaît donc comme 10^{18} fois plus « large » que l'intervalle unidimensionnel. En général, pour un espacement de 10^{-n} et un hypercube de dimension p, $[0, 1]^p$, la couverture est assurée avec 10^{np} points, soit un espace $10^{n(p-1)}$ fois plus « large » que l'intervalle unité.

Les problèmes à grande dimension sont souvent complexes, coûteux en temps de calculs et sont difficiles à utiliser dans des études comme l'optimisation ou l'inversion.
Face à ce problème et pour effectuer des traitements sur le modèle de référence dans des temps raisonnables, il existe deux méthodes usuelles :

- la réduction de la dimension,
- la métamodélisation.

La première consiste à représenter les données d'entrée dans un espace réduit de dimension $d' < d$. En optimisation par exemple, plus la dimension est grande, plus le domaine à explorer est grand, plus les temps de calculs sont importants. La réduction de la dimension permet donc de simplifier le problème et rendre les méthodes moins coûteuses. Dans certains cas, l'analyse de données, comme la régression ou la classification par exemple, peut donner des résultats plus précis sur un espace réduit que sur l'espace initial.
La seconde méthode, la méta-modélisation, consiste à remplacer le code de calcul coûteux par une fonction mathématique peu coûteuse à évaluer. Tous les traitements sont faits sur ce méta-modèle, fournissant des résultats approchés de ceux que nous aurions obtenus avec le vrai modèle.

Ces deux types de méthodes permettent de simplifier le problème, de le rendre moins coûteux en temps de calculs et contribuent à une meilleure compréhension du problème.

Ces méthodes seront l'objet du Chapitre II. Nous y explorerons les principales méthodes de réduction de la dimension et de méta-modélisation. Nous les utiliserons d'ailleurs successivement : nous réduirons la dimension du problème afin d'obtenir un meilleur méta-modèle. Nous

classerons ces différentes méthodes selon le but recherché et le problème étudié pour choisir les plus adaptées à notre problématique.

3.2 Etablir et appliquer une méthodologie de conception robuste

En avant-projets, l'objectif théorique est de répondre au plus juste aux spécifications. Dans un univers idéal, les performances d'un système sont invariantes quelles que soient les perturbations auxquelles il est soumis : c'est la vision théorique de la Figure I.16. Dans un univers réel, les performances d'un système sont soumises à de nombreuses perturbations se traduisant par des variations de ses performances : c'est la vision réelle de la Figure I.16. En raison de différentes variabilités, des systèmes issus d'une même conception ne possèdent jamais les mêmes performances. En dimensionnement, la prise en compte des variabilités potentielles sur les différents paramètres d'entrée fait que le système ne peut plus être vu comme déterministe. L'objectif de la méthodologie conception robuste est de réduire la sensibilité aux perturbations et de s'assurer que la probabilité d'atteindre les objectifs de performance représenté par le cercle vert est maximisée : c'est la vision robuste de la Figure I.16. Appliquée au dimensionnement en

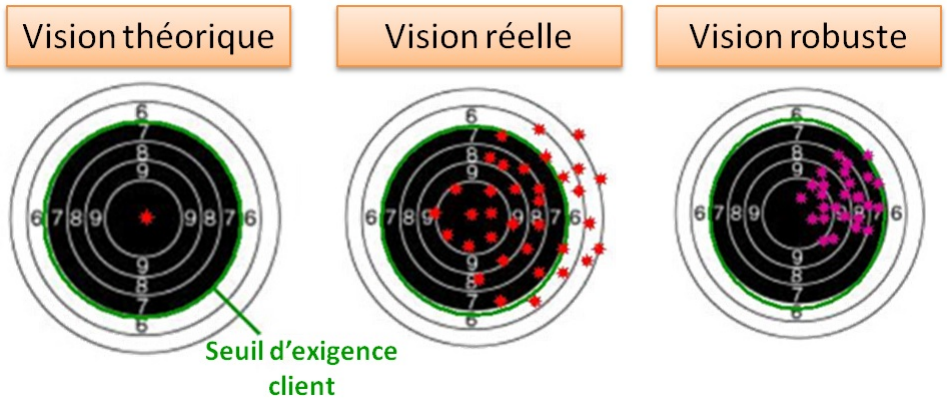

Figure I.16 – Illustration de la conception robuste (Snecma, 2012)

avant-projets, la conception robuste doit garantir une performance optimale selon des critères donnés mais aussi permettre d'assurer que la solution retenue sera stable par rapport :

- aux variations des paramètres de conception,
- aux changements de spécification,
- aux incertitudes liées aux méthodes de dimensionnement utilisées.

La prise en compte de ces variations permet de trouver l'optimum robuste du système, souvent différent de l'optimum global, comme le montre la Figure I.17. Dans l'approche déterministe, l'optimum global du système est évalué sans prise en compte d'un critère de robustesse, on

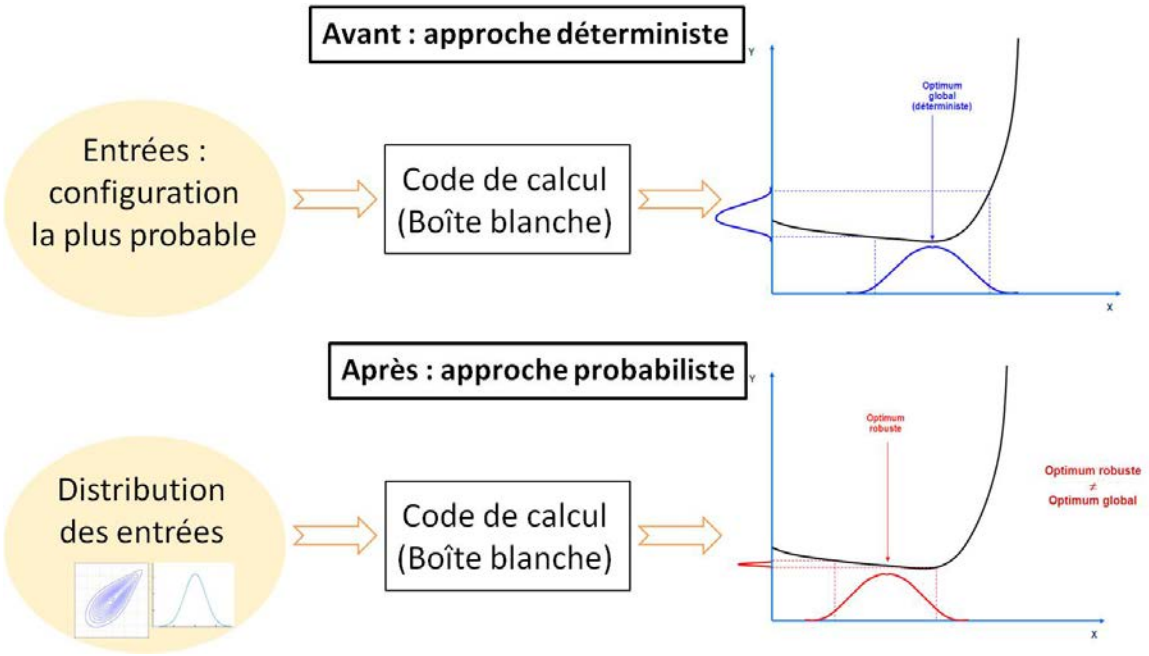

Figure I.17 – Conception robuste avant/après

choisit la meilleure configuration des entrées parmi toutes les configurations possibles sans considération des incertitudes. Dans une approche probabiliste, les variations des entrées sont prises en compte via des lois de probabilité. On obtient alors l'optimum robuste, moins bon en terme d'optimalité que l'optimum global mais beaucoup moins sensible aux variations possibles des entrées.
En pratique, il est possible d'effectuer une optimisation déterministe et d'évaluer ensuite la robustesse des optima trouvés afin de sélectionner celui qui convient le mieux à cet environnement incertain. Cependant, il se peut que certaines solutions robustes ne fassent pas partie des solutions déterministes trouvées. Il est donc souvent préférable d'introduire la robustesse dans le problème d'optimisation afin d'obtenir directement les solutions robustes. Le principal inconvénient de la prise en compte de la robustesse dans l'algorithme d'optimisation est le temps de calculs de la méthode. En effet, nous verrons que la méthode nécessite d'évaluer plusieurs fois la fonction à optimiser pour déterminer la robustesse en chaque point de l'optimisation.

Cet objectif sera l'objet du Chapitre III. Le but est de déterminer une méthodologie de résolution d'un problème d'optimisation robuste. En particulier, nous explorerons les différentes méthodes d'optimisation.

3.3 Résoudre des problèmes inverses mal posés

On reprend les notations du paragraphe 2.1 : on a le code de calculs f, les entrées sont notées x et la sortie y (et non plus X et Y) car ici on considère le problème comme étant déterministe. En effet, les méthodes que nous proposons pour résoudre un problème inverse ne prend pas en compte les incertitudes sur les entrées considérées.
Le problème direct (boîte blanche) consiste à trouver y connaissant x et f. Le problème inverse consiste à trouver x à partir d'observations de y et connaissant f.
Dans la majorité des cas, les problèmes inverses sont des problèmes mal posés, en opposition aux problèmes bien posés, notion définie par Hadamard en 1902. Il pensait que les modèles mathématiques représentant des phénomènes physiques devaient satisfaire trois propriétés :

1. une solution existe,
2. la solution est unique,
3. le comportement de la solution évolue continûment avec les conditions initiales.

Il n'est pas rare que l'existence de la solution ne soit pas assurée. Ensuite, il arrive souvent qu'un problème inverse ait plusieurs voire une infinité de solutions, surtout quand f est définie sur $\mathbb{R}^d, d > 1$, à valeurs dans $\mathbb{R}$. Deux exemples sont donnés à la Figure I.18. Enfin, la solution

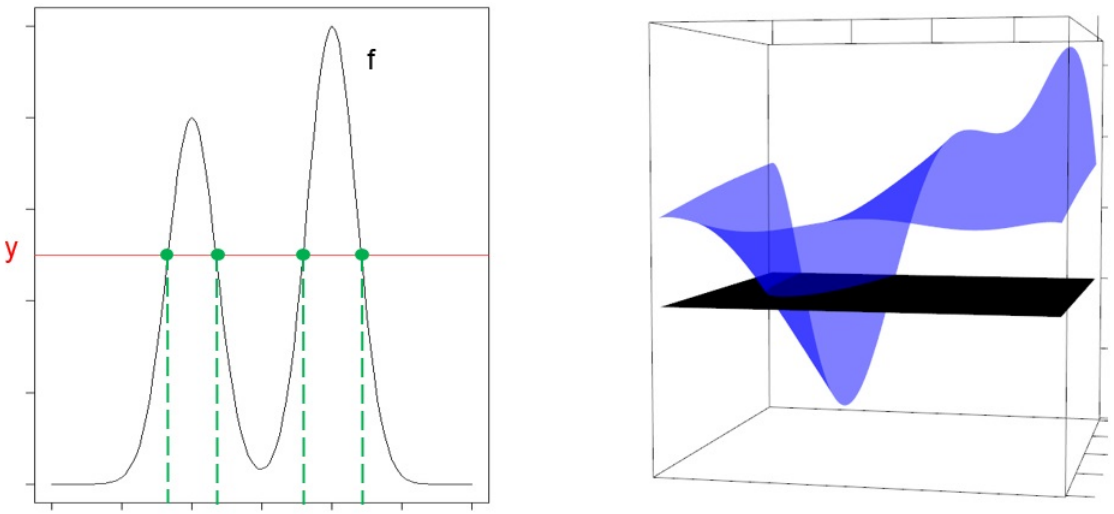

Figure I.18 – Exemples de problèmes avec plusieurs solutions : à gauche, une fonction réelle à 4 solutions ; à droite, une fonction bidimensionnelle avec une infinité de solutions formant une parabole

n'est pas forcément continue puisqu'elle peut être constituée de plusieurs composantes connexes disjointes par exemple (cf. Figure I.19). Peu de méthodes permettent de résoudre des problèmes inverses mal posés. La plupart d'entre elles passe d'abord par une étape de régularisation, qui consiste à rendre le problème bien posé, via l'introduction d'informations a priori sous forme

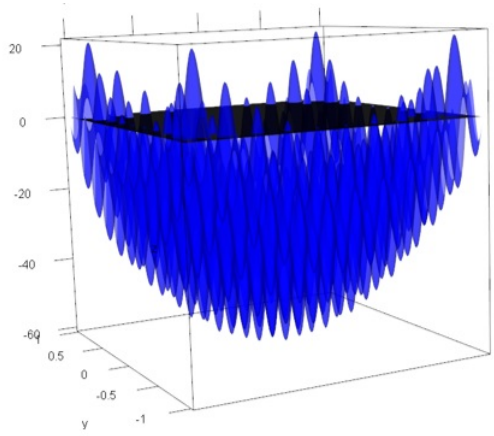

Figure I.19 – Exemple de problème inverse avec des solutions non continues : elles forment plusieurs composantes connexes disjointes

de pénalités.
Les deux méthodes que nous proposons dans cette thèse permettent de résoudre directement le problème mal posé, sans le modifier. Ces deux méthodes seront décrites dans le Chapitre IV.

4 Présentation des cas tests

Dans la thèse, nous avons utilisé deux cas tests. Le cas test principal représente la problématique de ce projet, à savoir améliorer et accélérer les études en avant-projet par la conception robuste et l'inversion de fonction. Ce cas test sera donc présent dans tous les chapitres de ce mémoire afin d'illustrer et de tester les méthodes qui seront décrites. Le cas test secondaire est un problème qui nous a été soumis en cours de thèse. Il permettra d'illustrer les limites de certaines méthodes.

4.1 Cas test principal : dimensionnement aéro-mécanique du CHP d'un turboréacteur

L'étude porte sur le dimensionnement aérodynamique et mécanique d'un compresseur haute pression (CHP) complet (rotor et stator) et du palier 3 avec le tourillon. Le fonctionnement de ces modules est décrit en section 1.2. Pour des raisons techniques, le dimensionnement thermodynamique ne fait pas partie du cas test. Nous connaissons le cycle thermodynamique du moteur étudié, nous le considérons donc comme fixé dans notre étude. Ce cas test nous permet d'obtenir une estimation de la masse du compresseur HP et de repérer la présence de problèmes d'intégration autour du palier 3 et du tourillon.

4.1.1 Outils utilisés

Le développement du modèle permettant le dimensionnement du CHP et du palier 3 a nécessité l'aide des ingénieurs en avant-projets mécaniques ainsi que l'utilisation d'outils existants tels que :

- l'outil de génération de veine pour le CHP,
- l'outil de dimensionnement mécanique de disque,
- l'outil de calcul de la masse des aubes,
- l'outil de dimensionnement du palier 3, du tourillon et de détection des problèmes d'intégration,
- l'outil de chaînage Optimus : outil permettant de regrouper plusieurs outils en les utilisant les uns à la suite des autres dans un même système.

Tous ces outils ont été développés en interne à Safran Aircraft Engine et reliés via le logiciel de chaînage, Optimus. Environnement de conception très utilisé par les ingénieurs, notamment en avant-projets, Optimus permet une visualisation et une exploration automatiques de l'espace de conception grâce à l'utilisation de « workflows ». Dans ces workflows, des graphes sont créés, représentant tout le système sous forme d'enchaînement de boîtes. Ce logiciel permet notamment de générer et d'exploiter des plans d'expériences mais aussi de mener des études d'optimisation sous contraintes.

Dans la suite, nous nous intéresserons à des méthodes développées durant cette thèse et appliquées à ce cas test. Ces méthodes ont été implémentées dans le logiciel R. Ce logiciel de programmation libre et gratuit est le plus souvent utilisé dans les traitements des données et l'analyse statistique. Les possibilités de ce logiciel sont d'autant plus importantes qu'il existe un nombre en croissance permanente de packages. L'ajout de ces librairies de fonctions permet d'effectuer certaines actions sans avoir à les coder à la main.

4.1.2 Hypothèses simplificatrices

Le dimensionnement du compresseur HP a été réduit au dimensionnement des aubes et des disques pour chaque étage, auxquels s'ajoutent les dimensionnements du palier 3, du tourillon et d'une partie du carter intermédiaire (carter structural assurant la rigidité du moteur et le transfert de poussée). Par souci de simplicité, toutes les autres pièces du CHP sont négligées. Dans chacun des outils de dimensionnement, les pièces du CHP sont vues comme des géométries régulières définies par un nombre fini de points (cf Figure I.20). Une aube peut être caractérisée par quatre points. De même, les disques, le palier 3 et le tourillon sont représentés de manière

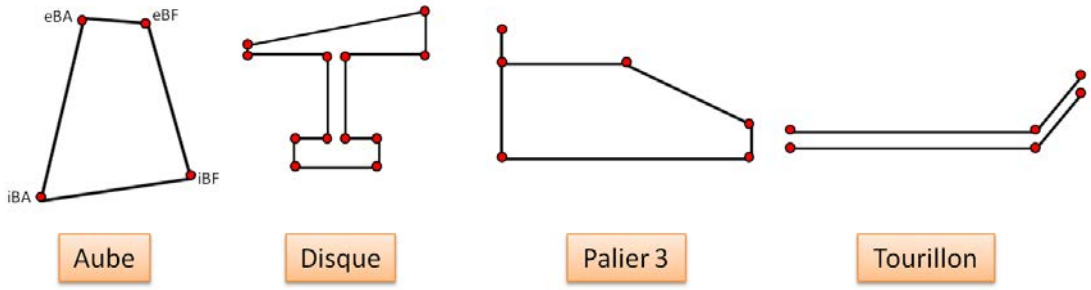

Figure I.20 – Schématisation des éléments du CHP

simplifiée. En plus de ces simplifications géométriques, des hypothèses simplificatrices et des calculs approchés ont été utilisés à chaque étape du dimensionnement afin de rendre le cas test facilement et rapidement utilisable et exécutable. Ces hypothèses et choix de calculs approchés sont soit inhérents à chaque outil préexistant et fourni pour le cas test, soit ont été décidés avec les ingénieurs avant-projets et les experts métiers.
En entrée de notre système, on dispose de deux types de données :

- les données de cycle,
- les paramètres utiles pour le dimensionnement du palier 3 et du tourillon.

Certaines données ont été fixées au début du projet afin d'éviter au maximum les erreurs de compilation du code et réduire la complexité du problème.

4.1.3 Chaînage du cas test dans l'outil Optimus

Le modèle peut être représenté comme une chaîne, établie dans le logiciel de chaînage Optimus. On parle alors d'un « workflow ». Le workflow général doit effectuer les différents dimensionnements selon le schéma de la Figure I.21. Les résultats du dimensionnement aérodynamique permettent de réaliser le dimensionnement mécanique qui consiste à calculer la masse des aubes et la force centrifuge ainsi qu'à dimensionner les disques, ceci pour chaque étage. Le dimensionnement mécanique fournit alors la masse totale du CHP. C'est ce dimensionnement qui, avec les résultats du dimensionnement du palier 3 et du tourillon, permet de déterminer les clashs.
La Figure I.22 est une illustration du workflow Optimus. Les boîtes bleues sont les entrées de la chaîne, les boîtes rouges en sont les sorties. Les boîtes oranges sont les boîtes d'action. Elles ont besoin des boîtes vertes qui les entourent afin de pré- et post-traiter les fichiers nécessaires à l'action.
Pour faire un rapprochement avec les sections précédentes, on peut préciser que la première ligne de la chaîne correspond à la boîte aérodynamique et les deux autres lignes à la boîte

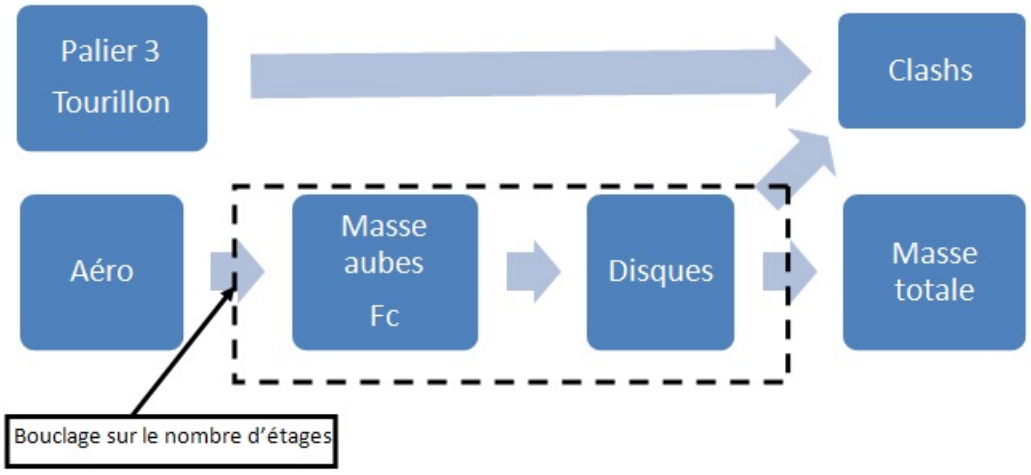

Figure I.21 – Schéma général du dimensionnement aéro-mécanique pour le compresseur HP

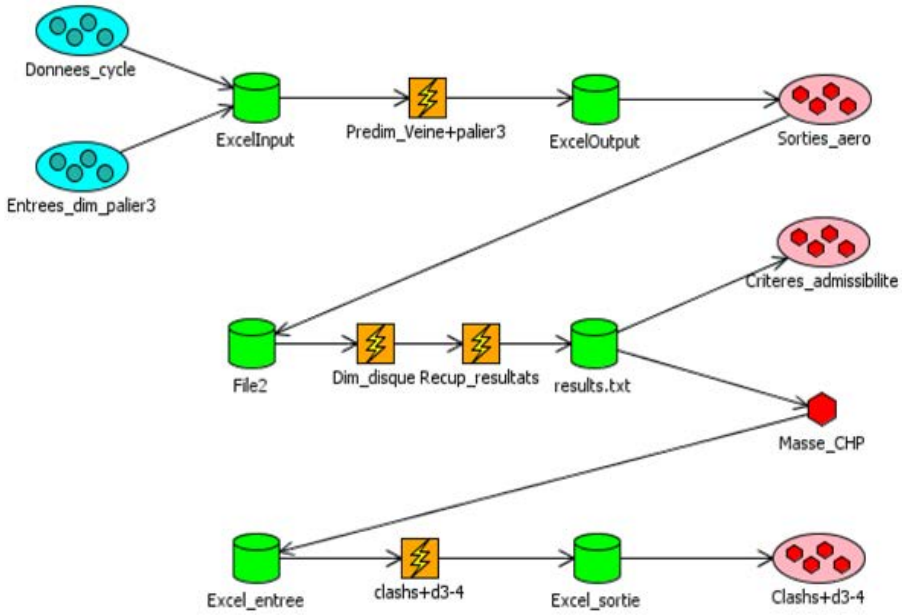

Figure I.22 – Chaînage aéro-mécanique pour le dimensionnement du CHP

mécanique.

Dans notre cas, chaque boîte d'action correspond à l'appel d'un des outils de dimensionnement énumérés précédemment. En entrée, on retrouve les données de cycle et les paramètres géométriques pour le dimensionnement du palier 3 et du tourillon.

La première action fait appel à l'outil de génération de veine et à celui de dimensionnement du palier 3 et du tourillon. En sortie de cette première action, on récupère les données de veine utiles pour la suite de la chaîne, à savoir la longueur totale du CHP et son nombre d'étages. Le

reste des résultats, c'est-à-dire le dimensionnement des aubes, du palier 3 et du tourillon, sont disponibles dans les fichiers de sortie de chaque outil.
La deuxième action (deuxième ligne) fait appel à un autre workflow Optimus. Ce second workflow est constitué de deux actions :

1. appel à l'outil de calcul de la masse des aubes,
2. appel à l'outil de dimensionnement des disques.

Ce workflow secondaire réalise le dimensionnement du disque et le calcul de la masse des aubes pour un étage. On récupère en sortie la masse de l'étage et des critères d'admissibilité pour le disque dimensionné (cinq variables de sortie qui doivent être positives). Ce workflow est appelé autant de fois qu'il y a d'étages dans le CHP.
La dernière action (troisième ligne) fait appel à l'outil de détection d'un problème d'intégration du palier 3 et du tourillon sous le CHP. Cet outil a besoin du dimensionnement du premier disque ainsi que de la veine. A partir des données obtenues lors des deux actions précédentes, cinq variables sont calculées. Appelées « clashs », ce sont les variables de sortie de cette dernière action. Elles représentent la présence (ou non) d'un problème d'intégration à un endroit précis du palier 3 ou du tourillon (cf. Figure I.23). Il s'agit des clashs entre :

1. l'attache du palier 3 et la veine,
2. le coin droit haut du palier 3 et la veine,
3. le coin droit du palier 3 et le disque,
4. le tourillon et le palier 3,
5. le tourillon et le disque.

Les contraintes de clashs correspondent au fait que ces sorties sont positives (ce qui traduit qu'il n'y a pas de problème d'intégration).

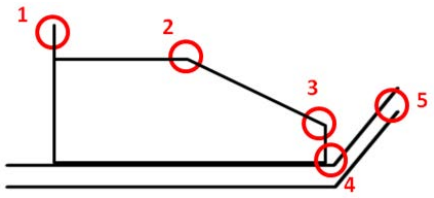

Figure I.23 – Zones possibles de clashs sur le palier 3 et le tourillon

Un tel modèle nous permet d'obtenir un dimensionnement aéro-mécanique d'un CHP en deux minutes, ce qui est relativement rapide.

Le lancement de l'expérience nominale de la chaîne fournit le résultat graphique de la Figure I.24. En vert, ce sont les aubes des roues mobiles (le premier élément vert représente le

carter primaire), en bleu celle des roues fixes, en rose les disques des roues mobiles, en rouge le palier 3, en violet le tourillon CHP et en marron l'axe moteur.

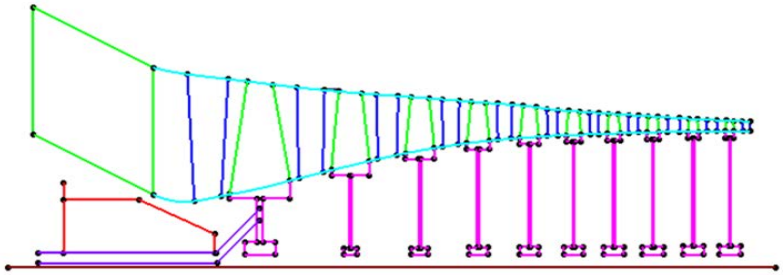

Figure I.24 – Représentation 2D du dimensionnement du CHP obtenu par le modèle

La représentation obtenue a été validée, en partenariat avec un ingénieur avant-projets, par comparaison avec la coupe réelle du moteur étudié. L'aube de la première roue mobile est bien dimensionnée et l'écart radial maximal pour le reste des aubes ne dépasse pas 8 mm. L'encombrement du palier 3 est bien représenté. La représentation du premier disque est proche de celui de la coupe moteur. La représentation des disques suivants importe peu puisqu'ils n'entrent pas en compte dans le problème d'intégration. En outre, la masse totale obtenue est d'un ordre de grandeur cohérent avec la masse réelle du CHP, pour le moteur considéré.

On dispose donc, en entrée de notre système, de données de cycle, dont certaines sont fixées, et de données géométriques figées. En sortie, nous avons la masse du CHP, sa longueur totale et son nombre d'étages. Nous disposons également de nombreuses contraintes en sortie : un vecteur de contraintes géométriques et un vecteur de clashs.

4.2 Cas test secondaire : calibration des jauges de support palier et exploitation

Dans ce cas test, nous ne sommes plus du tout en phase avant-projets puisqu'il s'agit d'essais physiques menés sur des pièces seules ou en situation dans le moteur. Le but est de récupérer les données d'essais menés sur un support palier et d'en extraire un modèle permettant l'extrapolation. L'exemple traité ici porte sur l'étude du support palier 1 qui peut se trouver en butée de l'arbre BP. Vu sur une coupe moteur, le support palier peut être représenté comme un cône sur lequel on va fixer les différentes jauges, comme le montre la Figure I.25. Le chargement est appliqué au palier. Dans les essais disponibles ici, nous disposons de quatre jauges équi-réparties sur le support palier, que nous représentons par un cercle selon une vue de face du moteur. Les

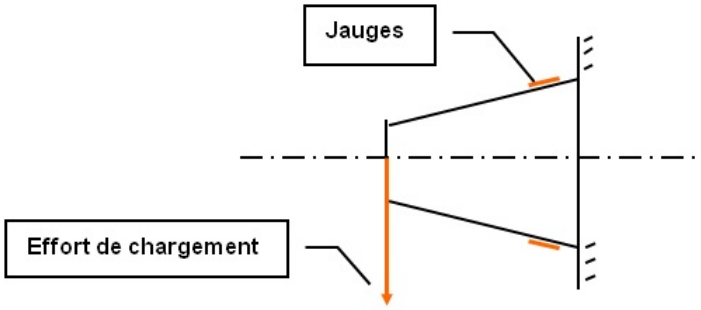

Figure I.25 – Représentation du support palier 1 avec la position des jauges (Snecma, 2014)

jauges, notées $S15$, $S30$, $S45$ et $S60$, sont placées respectivement à 90, 180, 270 et 0°, comme le montre la Figure I.26.

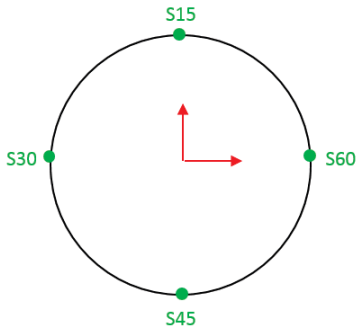

Figure I.26 – Représentation du support palier 1 avec ses 4 jauges

Ce palier est jugé axisymétrique, les jauges sont donc interchangeables. En d'autres termes, les combinaisons formées par la position de l'effort et la position des jauges sont identiques par révolution autour de l'axe moteur. Chacune des jauges mesure le taux de déformation subi par le palier. Cette déformation dépend de la force appliquée et l'angle de cette force par rapport au support palier.

L'étude de ce cas test est décomposé en deux parties :

1. calibration des jauges de support palier 1 par des essais statiques,
2. évaluation d'un effort à partir des signaux de jauges.

Nous allons voir en quoi consiste ces deux parties.

4.2.1 Calibration des jauges de support palier 1

Il s'agit d'une phase de calibration expérimentale où les essais effectués sont des essais statiques. On effectue une mesure des signaux de jauges pour une intensité (on parlera aussi de

force) et une direction de l'effort connues. Chacune des pièces étant symétrique de révolution, le niveau mesuré sur une jauge ne dépend, à intensité fixée, que de l'angle entre l'effort et la position angulaire de la jauge, grandeur que nous appelons la phase. On détermine ainsi, sur l'ensemble des jauges placées sur la circonférence de la pièce, la répartition des niveaux de déformation en fonction de la phase de l'effort. Cette répartition représente ce que l'on appelle la fonction de forme. On effectue cette même mesure pour différentes orientations de l'effort, ce qui permet d'enrichir l'échantillon et donc d'affiner la fonction de forme (correction de l'imprécision du positionnement des jauges par duplication des mesures). Toutes ces mesures sont effectuées pour différentes intensités de l'effort afin de déterminer une fonction permettant d'exprimer la déformation subie par la pièce, en fonction de la force et de l'angle. Nous l'appellerons la fonction de transfert.
Une fois la fonction de transfert déterminée, il est possible de l'utiliser pour réaliser la recombinaison d'effort à partir de signaux de jauges mesurés lors d'un essai machine ou moteur. En théorie, la calibration doit être menée pour une gamme d'effort correspondant à celle attendue pendant cet essai, dans la limite d'endommagement des structures. Dans la pratique, on a des efforts lors de la calibration trois fois plus faibles que ceux attendus pour un essai de perte d'aube. Cette particularité sera prise en compte dans la détermination de la fonction de transfert qui sera traitée dans le chapitre II.

4.2.2 Évaluer un effort à partir des signaux de jauges

Dans cette deuxième partie, nous disposons de mesures acquises lors d'un essai machine ou moteur, pendant un temps défini, correspondant par exemple à une perte d'aube. On cherche alors à positionner la fonction de transfert pour passer au plus près de ces valeurs en jouant sur deux paramètres que l'on cherche à déterminer :

- l'amplitude de l'effort,
- la phase de l'effort.

Ceci ressemble à la résolution d'un problème inverse dans lequel, connaissant la sortie (déformation), nous recherchons à déterminer les entrées de la fonction (force et angle). Cette partie sera donc traitée dans le chapitre IV.

5 Problématique : améliorer et accélérer les études en avant-projets par la conception robuste et les méthodes d'inversion

Le principal objectif en avant-projets est de réaliser rapidement le dimensionnement d'un nouveau moteur. Ce dimensionnement doit satisfaire les spécifications des avionneurs, les différentes contraintes de conception et les exigences de performances vis-à-vis de la concurrence ou des moteurs précédents.
La plupart des exigences à satisfaire en avant-projets sont traitées par optimisation sous contraintes. Ces méthodes d'optimisation permettent de réaliser une grande partie des études de dimensionnement et de répondre à de nombreux critères nécessaires à l'acceptation du projet par l'avionneur. Parmi ces critères se trouve la masse du moteur. Il a été expliqué dans la Section 1.3 l'importance de la masse en avant-projets. La masse annoncée en avant-projets représente un enjeu important et ne doit pas trop différer une fois le moteur conçu. C'est pourquoi des études d'optimisation robuste sont menées : il est essentiel que la masse annoncée soit accompagnée d'un intervalle de confiance et soit résistante à des variations des variables d'entrée. Ainsi, des petites variations de certaines variables géométriques entre la phase avant-projets et la phase de conception auront des conséquences minimes sur la masse, qui restera donc dans son intervalle de confiance.
Une autre exigence en avant-projets est le temps d'obtention du dimensionnement final. Comme cela a été expliqué dans la Section I.2, cette phase du développement d'un nouveau moteur peut être coûteuse en ressources informatiques et humaines. En effet, la phase avant-projets est caractérisée par de nombreux rebouclages entre différents métiers afin d'aboutir à un dimensionnement satisfaisant toutes les contraintes. Dans le but de faciliter, voire d'éviter certains de ces rebouclages, nous nous intéressons à la résolution de problèmes inverses. Ainsi, lorsqu'un corps de métier se retrouve face à un problème nécessitant un rebouclage, il est en mesure, grâce aux méthodes d'inversion, soit de résoudre seul le problème, soit de reboucler avec le corps de métier amont avec des propositions de modifications. Les avantages de ces méthodes par rapport à une optimisation sous contraintes sont les suivants :

- elles fournissent plusieurs solutions variées, ce qui permet de faire plusieurs propositions à tester,
- l'optimisation peut ne pas converger dans un temps raisonnable ou converger vers un optimum local non satisfaisant.

A terme, ces méthodes sont destinées à être implémentées dans les outils de dimensionnement

afin d'être utilisées de manière interactive. Ainsi, les problèmes seront détectés et résolus au sein même de l'outil.
La contrainte du temps peut également être traitée via les méthodes de réduction de la dimension et de méta-modélisation, comme nous l'avons décrit à la Section 3.1.

Les méthodes d'optimisation robuste et de résolution de problèmes inverses sont appliquées dans cette thèse au cas d'un compresseur HP mais peuvent être transposées à toute autre pièce du moteur. L'optimisation robuste portera sur la masse du compresseur et se fera sous contraintes. La résolution de problèmes inverses se fera sur le clash n°3 (cf. Figure I.23) : il y a un problème d'intégration et on tente de le résoudre en modifiant la veine aérodynamique.

Chapitre II

Réduction de la dimension et méta-modélisation

Dans ce chapitre, nous présentons différentes méthodes de réduction de la dimension et de méta-modélisation. Dans les deux cas, les méthodes sont très nombreuses et variées. L'intérêt de ce chapitre n'est pas de toutes les décrire mais de les lister de manière la plus exhaustive possible puis de décrire uniquement celles qui ont été ensuite utilisées sur les cas tests. Une partie sera donc consacrée à la réduction de la dimension, la suivante à la méta-modélisation. Certaines méthodes de méta-modélisation sont également présentées en annexe B car elles peuvent être utilisées dans d'autres cas d'applications industrielles. Dans tous les cas, il est nécessaire de posséder un ensemble d'observations, appelé base d'apprentissage, qui doit être structuré. On parle alors de plan d'expériences. Cette notion fera l'objet de la deuxième section, la première étant consacrée à la normalisation des données.
Dans ce chapitre, nous ne nous intéressons qu'à une seule sortie d'intérêt à la fois. Nous voulons trouver un ensemble réduit d'entrées (obtenu par la réduction de la dimension) qui permette ensuite d'établir une représentation fidèle de la relation entre les entrées et la sortie d'intérêt (obtenue par la méta-modélisation).

Les méthodes de réduction de la dimension puis de méta-modélisation seront appliquées au cas test principal de dimensionnement du CHP. Les sorties d'intérêt sont la masse du CHP et la variable de clash (cf. Chapitre I, Section 4.1).
Les méthodes de méta-modélisation seront également appliquées au cas test secondaire (cf. Chapitre I, Section 4.2) pour lequel la calibration de jauges consiste à établir un modèle pour les déformations enregistrées lors d'essais statiques.

1 Normalisation des données d'entrée

En analyse de données, la normalisation permet de rendre toutes les entrées d'un système comparables en supprimant leur unité respective. Dans les cas de la réduction de la dimension et de la méta-modélisation, elle est indispensable car l'incompatibilité des mesures peut affecter les résultats et leur interprétation. En effet, des entrées exprimées en millimètres risquent d'avoir des valeurs beaucoup plus grandes que celles exprimées en mètres par exemple. Dans un méta-modèle, les premières peuvent être désignées comme très significatives comparées aux secondes du fait de leurs valeurs plus élevées. Mais ramenées à la même unité, la significativité peut s'inverser, conduisant à une toute autre conclusion.
Pour cela, une transformation est appliquée sur chaque entrée $x_1, \ldots, x_d$. Sous la condition que ces entrées soient bornées (pour tout $i = 1, \ldots, d$, $m_i \leq x_i \leq M_i$), la normalisation se présente de la manière suivante

$$x_i^{norm} = \frac{2x_i - m_i - M_i}{M_i - m_i}. \tag{II.1}$$

Cette notion ne doit pas être confondue avec la standardisation, qui, en statistiques, consiste à centrer et réduire une variable. Dans ce cas, il faut donc connaître la moyenne et l'écart-type de la variable puisque chaque valeur sera transformée en soustrayant sa moyenne et en divisant par son écart-type. Cette transformation permet ainsi de comparer plus facilement des distributions.
Il n'apparaît donc pas adapté d'utiliser la standardisation dans un cas déterministe, où les entrées n'ont pas de loi de probabilité, elles sont seulement déterminées par une unité et un intervalle de variation.

2 Plans d'expériences pour la réduction de la dimension et la méta-modélisation

La méthode des plans d'expériences consiste à effectuer une campagne de calculs la plus économique possible (le moins d'expériences possible) tout en assurant des résultats les plus précis possibles. Selon le type de plan choisi, la méthode des plans d'expériences permet

- de quantifier et de hiérarchiser l'influence de plusieurs facteurs sur la réponse,
- de modéliser le comportement du système.

2.1 Vocabulaire lié aux plans d'expériences

Les plans d'expériences s'appliquent sur des systèmes physiques réels ou modélisés, constitués d'entrées et de sorties. En théorie des plans d'expériences, les sorties sont nommées les réponses du système. Un PE peut être fait pour plusieurs réponses mais leur utilisation se fera pour chaque réponse une à une. Les entrées $x = (x_1, \ldots, x_d)$ sont les facteurs. Elles sont bornées, normalisées et discrétisées en plusieurs valeurs nommées niveaux. Chaque niveau se situe dans les bornes de variation du facteur. Par exemple, un facteur à deux niveaux prendra les valeurs -1 et 1, un facteur à trois niveaux les valeurs -1, 0 et 1.
Une expérience correspond à un calcul, c'est-à-dire un jeu des données d'entrée, auquel est associé la valeur de la sortie, obtenue par évaluation du code. Lorsque le système utilise justement un code de calcul, alors il n'y a pas de variabilité de mesure. En effet, un même jeu de données d'entrée donnera toujours la même valeur en sortie du code.
On nomme l'« effet » la quantité représentant l'influence d'un facteur sur la réponse. On dit que deux ou plusieurs facteurs ont une interaction si l'effet de chacun sur la réponse dépend du niveau pris par les autres. Le nombre de facteurs concernés fournit l'ordre de l'interaction. La Figure II.1 fournit un exemple d'interaction d'ordre 2 pour deux facteurs A et B à deux niveaux chacun. Dans les deux graphiques, chaque point correspond à la valeur de la réponse, pour un certain niveau de A et B.

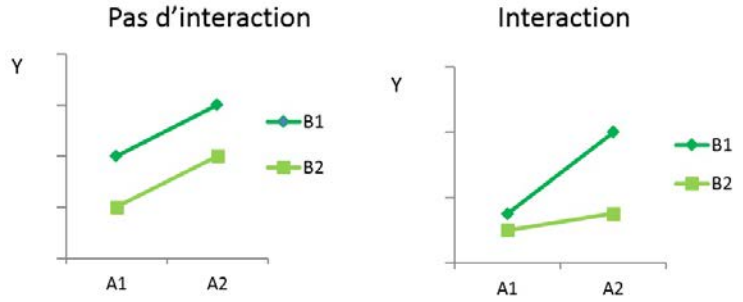

Figure II.1 – Exemple d'une interaction d'ordre 2 pour deux facteurs à deux niveaux

Une interaction est représentée comme la multiplication des facteurs concernés. Par exemple, l'interaction entre les facteurs A et B est notée AB. En pratique, les interactions d'ordre supérieur à 3 sont négligées. Dans la suite, le terme « interaction » sera utilisé pour désigner une interaction d'ordre 2.
Un alias est une confusion entre les effets. Par exemple, un facteur D est confondu avec l'interaction d'ordre 3, ABC, si D et ABC ont les mêmes expériences dans le plan. Ils prennent donc les mêmes niveaux. On note alors $D = ABC$. La résolution d'un plan est le nombre mi-

Confusion	Taille du plan	Résolution	Type de confusion	Effet des facteurs	Interactions
Gênantes	Faible	III	$A = BC$	Existence de confusions	Confondues avec les effets des facteurs
↑	↓	IV	$AC = DE$ $A = BCD$	Estimation possible	Existence de confusions
Pas gênantes	Grande	V	$AC = EFG$ $A = BCDE$	Estimation possible	Estimation possible

Tableau II.1 – Résolution d'un plan suivant les confusions

nimum de facteurs impliqués dans un même alias. Cette notion résume la structure d'un plan puisque plus la résolution est faible, plus le nombre d'expériences diminue, plus les confusions sont gênantes pour estimer les effets. Ceci est représenté dans le Tableau II.1. Il existe deux types de plans d'expériences :

- les plans factoriels,
- les plans pour surface de réponse.

2.2 Plans factoriels

Les plans factoriels sont des plans discrets orthogonaux. Les facteurs sont discrétisés pour ne prendre qu'un nombre fini de valeurs, les niveaux. L'orthogonalité d'un plan se définit de la manière suivante

Définition 1 (Orthogonalité)**.** *Un plan d'expériences est orthogonal si :*

i. à chaque niveau d'un facteur, tous les niveaux de n'importe quel autre facteur lui sont associés le même nombre de fois dans le plan,

ii. chaque niveau de chaque facteur apparaît le même nombre de fois.

Exemple 1. *(Benoist et al., 1994) propose un plan factoriel à deux facteurs (cf. Tableau II.2), X_1 prend 3 niveaux et X_2 prend 2 niveaux.*

En pratique, dans ces plans, les facteurs prennent 2 à 5 niveaux. Il existe deux types de plans factoriels :

- les plans complets,
- les plans fractionnaires.

#	X_1	X_2
1	-1	-1
2	-1	1
3	0	-1
4	0	1
5	1	-1
6	1	1

Tableau II.2 – Exemple de plan complet pour 2 facteurs

Nombre de facteurs	Nombre d'essais
2	4
3	8
4	16
5	32
10	1024
20	1048576

Tableau II.3 – Nombre d'essai des plans complets suivant le nombre de facteurs à 2 niveaux

Les plans factoriels complets sont des plans orthogonaux dont les facteurs prennent 2 ou 3 niveaux et où toutes les combinaisons possibles entre tous les niveaux des facteurs sont évaluées. Le plan du Tableau II.2 est un plan complet de deux facteurs respectivement à 3 et 2 niveaux. Lorsqu'on dispose de d facteurs à 2 niveaux, le plan complet est noté 2^d. Pour 3 niveaux, on le note 3^d. Si on a d_1 facteurs à 2 niveaux et d_2 facteurs à 3 niveaux, on note le plan $2^{d_1}3^{d_2}$.
Ces plans permettent d'estimer les effets des facteurs et toutes les interactions mais conduisent rapidement à un nombre important d'expériences, ce qui peut rendre la méthode coûteuse. Le Tableau II.3 donne le nombre d'essais nécessaires pour effectuer un plan complet avec des facteurs à 2 niveaux.
En pratique, ces plans ne sont pas souvent utilisés à cause de ce grand nombre d'expériences. On utilise plutôt des plans fractionnaires qui représentent une fraction orthogonale du plan complet. Cette fraction consiste à ne garder que certaines expériences du plan complet tout en conservant la propriété d'orthogonalité (cf. Tableau II.2).

Exemple 2. *Le plan fractionnaire de la Figure II.2 (à droite) à 4 expériences pour 3 variables (x_1, x_2, x_3) à 2 niveaux ne conserve que quelques expériences du plan complet 2^3 (à gauche).*

Ces fractions s'obtiennent en créant des confusions dans le plan complet (cf. Tableau II.1). La taille de la fraction dépend des quantités que l'on veut estimer. Si les interactions sont négligeables, alors un plan de résolution III est suffisant. En règle générale, comme on n'a aucun a

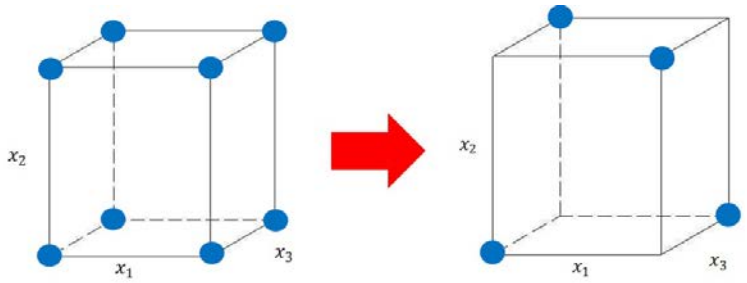

Figure II.2 – Sélection des expériences pour la réduction du plan complet 2^3

priori sur les interactions et qu'on ne souhaite pas les estimer, on choisit un plan de résolution IV. Sinon, on prend un plan de résolution V.

Parmi les plans fractionnaires, on trouve

- les plans de Plackett et Burman,
- les plans de Box,
- les plans de Taguchi.

Les plans de Plackett et Burman sont des plans restrictifs puisqu'ils ne permettent pas de prendre en compte les interactions. Ce sont donc des plans de résolution III où les effets des facteurs sont confondus avec les interactions. Le lecteur intéressé par ce type de plan pourra se référer à (Plackett and Burman, 1946).

Les plans de Box sont des plans plus récents. Notés $2^{d-k}, k < d$, correspondant au nombre d'expériences pour d facteurs à 2 niveaux, ces plans peuvent être de la résolution III pour les plus grands k à la résolution VII pour les plus petits k. (Box et al., 1978) ont établi un tableau des différents plans possibles pour des facteurs à 2 niveaux (jusqu'à 10 facteurs) avec les alias permettant de construire ces plans.

Enfin, les plans de Taguchi sont des plans fractionnaires orthogonaux tabulés. De nombreuses tables sont disponibles dans la littérature mais leur utilisation nécessite un minimum de connaissance sur leur fonctionnement. Le lecteur intéressé pourra se référer à (Sabre, 2007). Beaucoup utilisées en industrie, les tables de Taguchi ont été conçues par le statisticien Genichi Taguchi dans le but de minimiser l'effet des alias et des erreurs de mesure.

Dans un plan de Taguchi, les facteurs peuvent prendre un nombre de niveaux constant (tous les facteurs ont le même nombre de niveaux : 2, 3, 4, 5) ou mixte (2 et 3, 2 et 4).

Ces plans sont notés $L_g(m^n)$ ou $L_g(m^n m'^{n'})$ selon le cas constant ou le cas mixte, g étant le nombre d'expériences, m le nombre de niveaux par facteurs et n le nombre maximum de facteurs que le plan peut prendre en compte. Il existe deux types de tables de Taguchi :

- les tables de type 1 permettant l'identification des interactions,
- les tables de type 2 ne permettant pas l'identification des interactions.

Plus généraux que les plans de Box, les plans de Taguchi sont également plus faciles d'utilisation puisque les plans sont déjà construits. Le choix se fait par rapport au nombre de facteurs, au nombre de niveaux par facteur et à la résolution souhaitée.

2.3 Plans pour surface de réponse

Les plans étudiés dans la partie précédente permettent d'établir des modèles linéaires avec interactions, ce qui ne rend pas toujours compte correctement du phénomène étudié. Il existe en effet des situations où de tels modèles vont s'avérer trop pauvres, car la relation entre la sortie et les entrée présente un comportement quadratique voire non-linéaire. Afin de pallier ce type de problème, on utilise plutôt des plans pour surface de réponse. Il en existe plusieurs types :

- plan factoriel complet 3^d,
- plan de Box et Behnken,
- plan composite centré,
- plan D-optimal,
- plan « space-filling » (plans latins, plans maximin, etc).

Le but de ces plans est de modéliser avec précision la relation entre la réponse et les facteurs d'entrée. Ceux-ci sont quantitatifs, discrétisés de manière plus ou moins fine sur leur domaine de variation.

Le plan factoriel complet 3^d est le plan pour surface de réponse le plus simple et le plus naturel à imaginer. On a vu dans la partie sur les plans factoriels que ce plan pouvait être utilisé pour des facteurs discrets. Il peut également être utilisé pour des facteurs continus, à condition de coder les niveaux des facteurs par une base orthogonale de polynômes afin d'assurer l'orthogonalité du plan. Ce type de plans permet d'établir des modèles polynomiaux jusqu'à l'ordre 2 (termes quadratiques et interactions).

Les plans de Box et Behnken sont des fractions du plan complet 3^d. Ils permettent également d'estimer les coefficients de modèles quadratiques, mais avec moins d'expériences.

Les plans composites centrés sont constitués d'un plan factoriel complet ou fractionnaire à 2 niveaux, de points situé sur les axes des facteurs à une certaine distance du centre du do-

maine et d'expériences au centre du domaine.

Les plans D-optimaux font partie d'une famille plus grande appelée plans optimaux alphabétiques (du nom des critères d'optimalité remplis par les plans : A, D, E, I, etc). Ils sont particulièrement bien adaptés à des problématiques avec contraintes, telles que :

- contrainte du domaine de variation : essais impossibles,
- réutilisation d'essais : certains essais ont été réalisés au préalable mais pas au niveau des points expérimentaux préconisés par la théorie des plans d'expériences,
- contrainte sur le nombre d'essais maximal à effectuer : dans cette situation, la qualité du plan se dégrade, il y a notamment perte d'orthogonalité. La précision des estimateurs que l'on obtiendra sera beaucoup plus faible.

La question qui se pose est donc de trouver d'autres expériences dans ce domaine sous contraintes pour permettre une estimation optimisée de ce modèle : c'est l'objectif du plan D-optimal.
Ces trois types de plans sont étudiés dans (Myers and Montgomery, 2009) et (Pukelsheim, 1993).

Beaucoup utilisés dans le cas des essais numériques, les plans latins hypercubes permettent d'étudier des modèles complexes, non linéaires par exemple. Dans ce type de plan, chaque facteur prend beaucoup de niveaux équi-répartis sur son domaine de variation.

Définition 2. *Un plan latin hypercube (PLH) en N essais est un plan d'expériences pour lequel :*

i. chaque facteur a le même nombre de niveaux N,

ii. chaque facteur prend chaque niveau une fois et une seule. Les niveaux sont équi-répartis.

Ainsi, chaque colonne du plan d'expériences est un tirage aléatoire sans remise parmi $\{1, 2, \ldots, N\}$.

Un exemple d'un plan latin hypercube à 5 essais est donné à la Figure II.3. L'intervalle de variation de chaque facteur est séparé en 5 sous-intervalles (le nombre d'essais) et une valeur est tirée dans chaque sous-intervalle suivant une loi choisie (par défaut uniforme). Chaque facteur prend donc ici 5 niveaux équi-répartis entre son minimum et son maximum.
La propriété exigée pour un plan latin hypercube est la bonne répartition des points selon leur projection sur chacun des axes. Dans le but d'optimiser le plan choisi, trois autres propriétés peuvent être satisfaites :

- le remplissage : maximisation de la distance entre les 2 points les plus proches du plan (D_{min}),
- l'indépendance : maximisation du déterminant de la matrice de corrélation des paramètres (R),

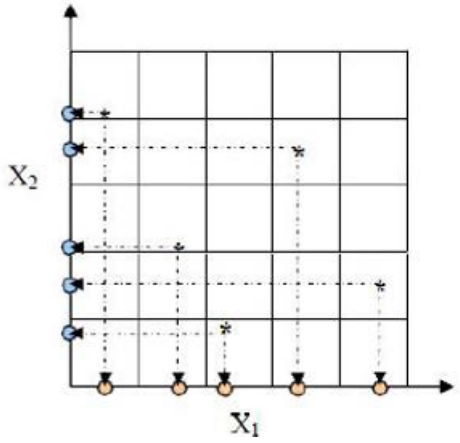

Figure II.3 – Plan latin hypercube pour 2 facteurs en 5 essais

- l'uniformité (discrépance) : minimisation de la distance à une répartition uniforme (CL_2).

En général, seules une ou deux propriétés sont choisies, par simplicité ou par incompatibilité. Il arrive souvent que le remplissage évolue de manière contradictoire avec la propriété de l'uniformité.

Sur la Figure II.4, le meilleur PLH est celui du milieu. En effet, pour celui de gauche, le remplissage et l'uniformité sont moins bons. Pour celui de droite, le remplissage est mauvais, l'indépendance n'est pas respectée et l'uniformité est beaucoup moins bonne.

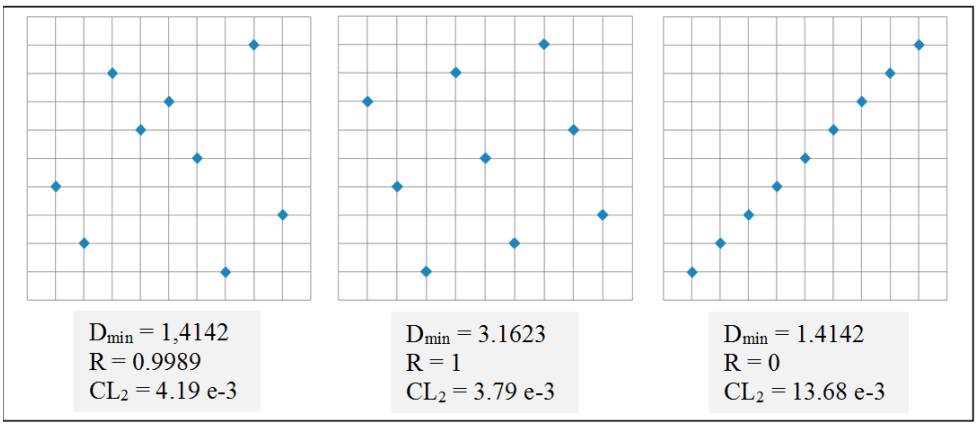

Figure II.4 – Comparaison de trois plans latins hypercubes pour illustrer les propriétés

En pratique, des méthodes algorithmiques comme le recuit simulé ou les algorithmes d'échange permettent de trouver, pour un problème donné, le meilleur plan latin hypercube, selon le critère retenu.

Le plan latin hypercube est surtout utilisé dans le cas de modélisation par krigeage, méthode d'interpolation spatiale nécessitant que le domaine d'entrée soit rempli le plus uniformément

possible. Cette méthode est présentée en Annexe B.

Une fois que nous avons l'échantillon d'apprentissage établi par le plan d'expériences choisi selon l'objectif, nous pouvons mettre en application les méthodes de réduction de la dimension et de méta-modélisation.

3 Méthodes de réduction de la dimension

Nous avons vu au Chapitre I ce que représentait le fléau de la dimension. Mener des études en grande dimension peut en effet devenir difficile autant du point de vue de la compréhension des résultats que des temps d'exécution pour obtenir ces résultats. Il est donc souvent nécessaire de réduire la dimension du problèmes avant de mener les études.
Différents classements des méthodes de réduction de la dimension sont proposés dans la littérature :

- application à la classification ou à la prédiction (distinction entre le domaine de l'apprentissage en analyse de données et celui de la statistique)
- méthodes de sélection de variables ou de transformation des données (aussi appelée extraction de fonctions) : classement proposé par (Cunningham, 2008) par exemple,
- méthodes linéaires ou non-linéaires : classement proposé par (Fodor, 2002) par exemple,
- le processus d'apprentissage est supervisé ou non-supervisé : distinction proposée par (Cunningham, 2008) par exemple.

La réduction par sélection de variables consiste à sélectionner les entrées les plus pertinentes sur la sortie parmi l'ensemble des entrées du système. La réduction basée sur la transformation des données consiste à exprimer les données sur un espace réduit par transformation linéaire ou non-linéaire des entrées. Ces méthodes, à l'image de la plus connue, l'ACP (Analyse en Composantes Principales), sont essentiellement basées sur les dépendances entre les entrées, et non entre les entrées et les sorties. Elles ne sont donc pas applicables ici puisque l'intérêt est porté sur les dépendances entre les entrées et les sorties. Les entrées sont supposées entre indépendantes deux à deux. Les méthodes étudiées dans cette section seront donc celles basées sur les dépendances entre les entrées et la sortie, comme la méta-modélisation. En effet, la construction de certains méta-modèles permet de réaliser une sélection de variables (nous le verrons d'ailleurs à la Section 3.2). Ces variables peuvent être les entrées du système mais aussi des combinaisons de ces entrées.
Dans cette partie, nous ne nous intéresserons qu'aux méthodes de sélection de variables. Le lec-

teur intéressé par les méthodes de transformation des données pourra se référer à (Cunningham, 2008) et (Fodor, 2002). Les méthodes de méta-modélisation seront expliquées à la Section 4.

Dans les problèmes de prédiction, la sortie est vue comme la variable à prédire et les entrées sont des variables prédictives, appelées aussi « prédicteurs ». Les méthodes de sélection de variables consistent à sélectionner un sous-ensemble de variables prédictives significatives (on parle de variables pertinentes en classification) par rapport à la variable à prédire.
Il existe différentes méthodes de sélection de variables pour la prédiction :

- les méthodes de criblage,
- les méthodes heuristiques de sélection de modèles,
- l'analyse de sensibilité.

Les méthodes de criblage sont des méthodes qualitatives permettant de traiter des problèmes en grande dimension (plusieurs dizaines, voire plusieurs centaines de prédicteurs). Elles sont basées sur une exploration rapide du code de calculs, qui peut être coûteux, par variation des entrées.
Les méthodes de sélection de modèles consistent à comparer les termes d'un modèle paramétrique pour ne garder que les termes significatifs. Ces méthodes sont d'autant plus utiles lorsque le but est de modéliser la relation entre une sortie et les entrées du système. Si la relation est assez régulière, la sélection de modèles permet de sélectionner les variables influentes et d'établir le modèle dans le même temps.
L'analyse de sensibilité repose sur l'estimation d'indices de sensibilité permettant de hiérarchiser les entrées selon la sensibilité de la sortie à leur variation. La particularité de cette méthode est qu'elle peut nécessiter un méta-modèle pré-établi qui doit présenter de bonnes qualités prédictives, si les calculs sont coûteux ou autorisés en un nombre limité.

Pour réaliser ces différentes méthodes de sélection de variables, il est nécessaire de posséder un ensemble d'observations, appelé base d'apprentissage, qui doit être structuré. Il s'agit du plan d'expériences décrit à la section précédente.
Les méthodes de criblage font appel à des plans factoriels. Pour les méthodes de sélection de modèle, on peut également utiliser ces modèles mais, comme nous l'avons vu à la section précédente, les plans « space-filling » tels que les plans latins hypercubes offrent davantage de possibilités. Ce seront ces plans que nous utiliserons pour établir un méta-modèle et calculer les indices de sensibilité.

3.1 Sélection par criblage

Certaines méthodes de criblage (screening en anglais) utilisent des plans factoriels fractionnaires pour des facteurs à 2 niveaux. Rappelons que si les interactions sont négligeables, alors on peut utiliser un plan de Plackett et Burman ou un plan de résolution III (Box ou Taguchi). Si au contraire on n'a aucun a priori sur les interactions, on choisira plutôt un plan de résolution au moins IV (Box ou Taguchi).

Remarque 1. *Une difficulté possible avec les codes de calculs est que certaines expériences ne sont pas exploitables puisqu'elles ne sont pas menées à terme. C'est notamment le cas lorsque certaines contraintes sont prises en compte directement dans l'outil. En effet, ces contraintes sont des équations dépendant des entrées. Si ces contraintes ne sont pas satisfaites, alors le calcul n'est pas effectué ou renvoie une erreur. On parle alors d'échec. Cependant, dans un plan factoriel, toutes les expériences doivent être utilisables afin d'assurer l'orthogonalité du plan et d'en tirer des résultats pour le criblage.*

Une fois le plan choisi et les expériences effectuées, il faut analyser les résultats du plan. Le but est d'estimer les effets des facteurs pour en dégager les plus influents. Cette sélection peut également être validée statistiquement. Les différentes étapes de l'analyse des résultats pour un plan factoriel sont :

- l'estimation des effets par des calculs simples de moyennes,
- la représentation des effets à l'aide de graphes d'effets,
- le test de la significativité de ces effets grâce à l'ANOVA.

On considère les N expériences $(x^1, y_1), \ldots, (x^N, y_N)$ où les x^i sont des vecteurs de taille d et y est un sortie scalaire.

Estimation des effets Grâce à la notion d'orthogonalité des plans factoriels, le calcul des effets se fait selon les mêmes règles, que l'on dispose d'un plan complet ou fractionnaire, de facteurs à deux niveaux ou plus. Pour ces calculs, nous donnons donc la formule générale, applicable pour deux niveaux ou plus, puis nous nous concentrons plus précisément sur le cas des plans à deux niveaux pour lesquels on dispose de règles supplémentaires très intéressantes.

L'effet global est la moyenne des valeurs de la sortie y sur tous les résultats d'essais. Cette moyenne correspond à la valeur de la réponse sans toucher aucun facteur. Grâce à l'orthogonalité du plan, c'est un bon estimateur de la moyenne de l'ensemble des combinaisons possibles.

Cette moyenne est simplement l'estimateur empirique suivant

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i. \tag{II.2}$$

Dans le cas d'un plan factoriel orthogonal, l'estimation des effets se fait par des calculs simples de moyenne. L'effet du facteur $x_j, 1 \leq j \leq d$ quand celui-ci se trouve au niveau k s'écrit :

$$e_j^k = \frac{1}{card\{y_i : x_j = k\}}\sum_{i=1}^{N} y_{i|x_j=k} - \bar{y} = \bar{y}_{x_j=k} - \bar{y}, \tag{II.3}$$

où $\bar{y}_{x_j=k}$ est la moyenne des y_i tels que la variable x_j est au niveau k. Dans le cas particulier des plans factoriels à deux niveaux, on a $k = +1$ ou $k = -1$, puisque les entrées sont normalisées et discrétisées. Les effets vérifient donc $e_j^{-1} = -e_j^{+1}$, pour tout $j = 1, \ldots, d$. D'où l'écriture de l'effet moyen du facteur x_j à 2 niveaux :

$$E_j = \frac{\bar{y}_{x_j=+1} - \bar{y}_{x_j=-1}}{2} = \frac{e_j^{+1} - e_j^{-1}}{2} \tag{II.4}$$

L'estimation des interactions se fait aussi par des calculs simples de moyenne. L'effet de l'interaction entre x_j et $x_{j'}$ quand le premier est au niveau k et le second au niveau k' s'écrit :

$$\begin{aligned} e_{jj'}^{kk'} &= \frac{1}{card\{y_i : x_j = k \text{ et } x_{j'} = k'\}}\sum_{i=1}^{N} y_{i|x_j=k \text{ et } x_{j'}=k'} - e_j^k - e_{j'}^{k'} - \bar{y} \\ &= \bar{y}_{x_j=k \text{ et } x_{j'}=k'} - \bar{y}_{x_j=k} - \bar{y}_{x_{j'}=k'} + \bar{y}. \end{aligned} \tag{II.5}$$

Dans le cas particulier des plans factoriels à deux niveaux, les effets vérifient les trois égalités suivantes :

$$\begin{aligned} e_{jj'}^{-1,+1} &= -e_{jj'}^{-1,-1}, \\ e_{jj'}^{+1,-1} &= -e_{jj'}^{-1,-1}, \\ e_{jj'}^{+1,+1} &= e_{jj'}^{-1,-1}. \end{aligned}$$

L'effet moyen de l'interaction $x_j x_{j'}$ s'écrit :

$$E_{jj'} = \frac{\bar{y}_{x_j x_{j'}=+1} - \bar{y}_{x_j x_{j'}=-1}}{2} \tag{II.6}$$

Graphes des effets Les graphes des effets (pour les facteurs et les interactions) permettent de visualiser clairement l'importance des différents facteurs et des différentes interactions sur la réponse. En effet, ils fournissent une représentation graphique des variations de la réponse quand un facteur change de niveau.
Le graphe des effets des facteurs est représenté par autant de graphiques qu'il y a de facteurs. La Figure II.5 donne un exemple de graphe des effets pour trois facteurs A, B et C obtenu avec le logiciel Minitab.

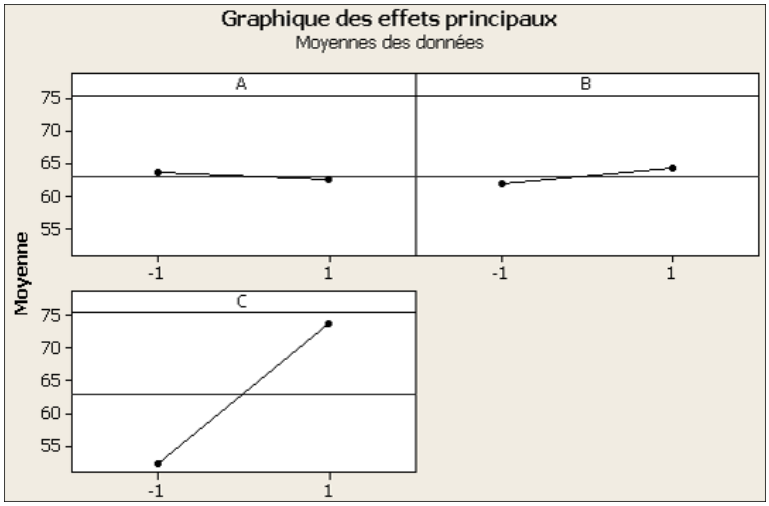

Figure II.5 – Exemple de graphe des effets pour des facteurs à 2 niveaux

La ligne horizontale représente l'effet global. Nous pouvons tirer plusieurs conclusions de ces graphes :

- Le facteur A a un effet négatif sur la réponse, les facteurs B et C ont des effets positifs.
- Le facteur C est celui qui a l'effet le plus important sur la réponse.
- Les facteurs A et B ont des effets pratiquement nuls (droites des effets proches de l'horizontale).

Le graphe des interactions est représenté par autant de graphiques qu'il y a d'interactions entre les facteurs. Pour d facteurs, il y a $\frac{d(d-1)}{2}$ interactions d'ordre 2. La Figure II.6 donne un exemple de graphe des interactions pour les trois facteurs A, B et C déjà considérés précédemment. On a trois interactions $A \times B$, $A \times C$ et $B \times C$. Nous pouvons tirer plusieurs conclusions de ces graphes :

- Il n'y a pas d'interaction entre A et B par rapport à la sortie puisque les deux droites sont parallèles.
- L'interaction entre B et C a peu d'effet sur la réponse.
- Seule l'interaction entre A et C semble influencer la réponse.

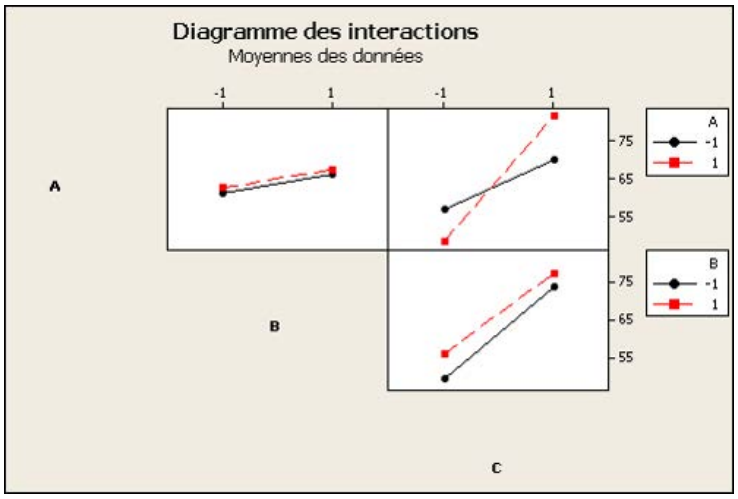

Figure II.6 – Exemple de graphe des interactions pour des facteurs à 2 niveaux

Test de la significativité des effets Afin de confirmer les conclusions obtenues avec les graphes précédents, il est courant de passer par une étape de validation statistique des effets. Cette étape consiste à tester la significativité des facteurs en comparant, pour la sortie, la variance totale à celle due à chaque facteur. Pour cela, nous disposons de deux tests :

- le test de Fisher pour l'analyse de la variance (ANOVA) teste la significativité globale d'un facteur.
- le test de Student teste la significativité individuelle de chacun des niveaux d'un facteur.

Pour des facteurs uniquement à 2 niveaux, les deux tests sont équivalents. Si certains facteurs ont 3 niveaux ou plus, alors le test de Student affine les résultats de l'analyse de la variance en examinant les effets de chacun des niveaux.

Pour les plans factoriels orthogonaux à N expériences, la méthode générale de l'analyse de la variance (ANOVA) repose sur le calcul des sommes de carrés suivants :

- la somme des carrés totale

$$\begin{aligned} SC_T &= \sum_{k=1}^{N} (y_k - \bar{y})^2 \\ &= SC_M + SC_R, \end{aligned} \tag{II.7}$$

où $\bar{y}$ est la moyenne de toutes les expériences, SC_M est la somme des carrés due aux facteurs et aux interactions (part expliquée de la variation de la réponse) et SC_R est la somme des carrés des résidus (part inexpliquée de la variation de la réponse), il s'agit de la décomposition ANOVA ;

- la somme des carrés des facteurs

$$SC_A = \frac{N}{n_A} \sum_{i=1}^{n_A} (e_A^i)^2, \tag{II.8}$$

où n_A est le nombre de niveaux du facteur A et e_A^i l'effet de A au niveau i ;

- la somme des carrés des interactions

$$SC_{AB} = \frac{N}{n_A n_B} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} (e_{AB}^{ij})^2, \tag{II.9}$$

où e_{AB}^{ij} est l'effet de l'interaction $A \times B$ pour la modalité (i, j).

La somme des carrés des résidus comprend la somme des carrés des interactions d'ordre supérieur à 2 ainsi que les combinaisons non évaluées du fait de la fraction du plan. Pour un plan complet par exemple, SC_R ne comprend que les sommes des carrés des interactions d'ordre 3 et plus. Dans le cas particulier du plan complet pour deux facteurs, qu'on ne peut pas fractionner, on a $SC_R = 0$. On dit que le plan est saturé. Dans ce cas particulier, on a montré en Annexe A l'égalité suivante

$$SC_T = SC_A + SC_B + SC_{AB} \tag{II.10}$$

La méthode ANOVA nécessite également le calcul du nombre de degrés de liberté :

- total : $ddl_T = N - 1$,
- d'un facteur : $ddl_A = n_A - 1$,
- d'une interaction : $ddl_{AB} = (n_A - 1)(n_B - 1)$,
- résiduel : $ddl_R = ddl_T - ddl_M$, où ddl_M est la somme des degrés de liberté de tous les facteurs et de toutes les interactions.

Les sommes des carrés et les degrés de liberté permettent de calculer les variances

- totale : $V_T = \frac{SC_T}{ddl_T}$,
- des facteurs : $V_A = \frac{SC_A}{ddl_A}$,
- des interactions : $V_{AB} = \frac{SC_{AB}}{ddl_{AB}}$,
- des résidus : $V_R = \frac{SC_R}{ddl_R}$.

Le test de Fisher consiste à tester l'hypothèse H_0 : « l'effet du facteur analysé est nul ». Pour un facteur A, la statistique de test est $F_A = \frac{V_A}{V_R}$. Sous H_0, F_A suit la loi de Fisher-Snedecor $\mathcal{F}(ddl_A, ddl_R)$. On note $F_{1-\alpha;ddl_A:ddl_R}$ le quantile d'ordre $1-\alpha$ de la loi de Fisher-Snedecor (cf. Figure II.7). La zone de rejet de l'hypothèse H_0 est $\{F_A > F_{1-\alpha;ddl_A;ddl_R}\}$.

La p-value mesure à quel point les données plaident contre l'hypothèse nulle. Il s'agit de la probabilité, pour une quantité observée f_A de F_A, d'obtenir un résultat égal ou plus extrême

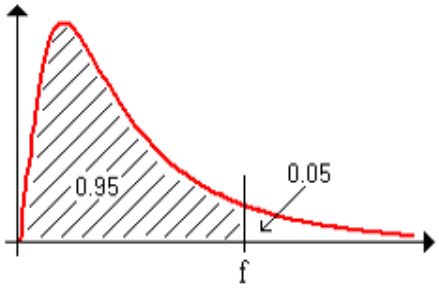

Figure II.7 – Loi de Fisher-Snedecor et son quantile d'ordre 0.95 ($\alpha = 5\%$)

que f_A sachant que H_0 est vraie : $\mathbb{P}(F_A \geq f_A | H_0)$. Les seuils suivants sont généralement pris pour référence :

- $p \leq 0.01$: très forte présomption contre l'hypothèse nulle,
- $0.01 < p \leq 0.05$: forte présomption contre l'hypothèse nulle,
- $0.05 < p \leq 0.1$: faible présomption contre l'hypothèse nulle,
- $p > 0.1$: pas de présomption contre l'hypothèse nulle.

Ainsi, plus la p-value est petite pour un facteur, plus la significativité de l'effet du facteur est grande. En faisant cela pour chaque facteur, on effectue une sélection de variables puisqu'on éliminera de l'étude les facteurs les moins significatifs en les fixant à une valeur nominale dans le code de calculs.

Le test de Student ne sera pas détaillé ici puisque les études de criblage utilisent principalement des plans à 2 niveaux. Nous l'expliquons tout de même en Annexe A, car il peut parfois être intéressant de tester la significativité de termes quadratiques par exemple ou la non-linéarité de la sortie suivant certains facteurs, en utilisant des plans à plus de 2 niveaux.

Pour plus d'explications sur les méthodes de criblage, le lecteur pourra se référer à (Sergent et al., 2009).
Ces méthodes sont très utiles pour effectuer une sélection rapide des facteurs les plus influents sur la réponse, surtout lorsqu'ils sont nombreux. Dans un objectif de méta-modélisation, on trouve une autre méthode de sélection de variables : la sélection de modèles.

3.2 Sélection de modèles paramétriques

Lorsque le but est d'établir ensuite un méta-modèle et que la sortie semble avoir un comportement plutôt régulier, il est judicieux de choisir un méta-modèle polynômial et d'effectuer la sélection de variables directement dans la construction du méta-modèle.

Il est essentiel de disposer ici, comme pour établir n'importe quel méta-modèle d'ailleurs, d'un échantillon d'apprentissage pour construire le modèle et d'un échantillon de test pour vérifier les qualités prédictives du modèle et donc le valider. L'échantillon d'apprentissage sera composé de N observations $(x^1, y_1), \ldots, (x^N, y_N)$, où les x^i sont des vecteurs de taille d puisqu'on considère avoir d entrées, les y_i sont des observations de la sortie scalaire. L'échantillon de test comportera M observations, $M < N$. On choisit souvent $M = \left[\frac{N}{3}\right]$, où [.] désigne la fonction partie entière.
De nombreuses méthodes existent afin de choisir le polynôme le plus simple, ajustant bien les données (bonne estimation) de l'échantillon d'apprentissage et conduisant à de bonnes prédictions pour l'échantillon de test. On parlera d'un modèle parcimonieux (ou efficace). Pour la suite, il est essentiel de fixer un modèle minimal (ou nul) qui comporte uniquement une constante et un modèle maximal (ou complet) choisi selon la complexité maximale acceptée pour le modèle. On considérera que le modèle complet comporte p termes.

On note les sommes de carrés suivantes, calculées sur l'échantillon d'apprentissage :
- sur les résidus : $SSE = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$,
- totale : $SST = \sum_{i=1}^{N}(y_i - \bar{y})^2$,
- de la régression : $SSR = \sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2$,

où les y_i sont les valeurs de l'échantillon d'apprentissage et calculées via le code de calculs, $\hat{y}_i$ les valeurs de y obtenues grâce au modèle choisi et $\bar{y}$ la moyenne des observations de l'échantillon d'apprentissage.
On a bien sûr $SST = SSR + SSE$.

3.2.1 Critères de sélection de modèles

De nombreux critères de choix de modèle sont présentés dans la littérature sur la régression linéaire multiple. Le choix du critère est déterminant lorsqu'il s'agit de comparer des modèles de niveaux différents. En pratique, les plus utilisés ou ceux généralement fournis par les logiciels sont les suivants :

Coefficient de détermination On appelle coefficient de détermination R^2 calculé sur l'échantillon d'apprentissage le rapport

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST},$$

qui est la part de variance de la sortie expliquée par le modèle de régression. Ce coefficient est un indice de qualité du modèle mais qui a la propriété d'être croissant en fonction du nombre de termes dans le modèle. Il ne peut donc servir qu'à comparer deux modèles de même niveau, c'est-à-dire avec le même nombre de termes.

Coefficient de détermination ajusté Le coefficient R^2 ajusté introduit une pénalisation liée au nombre de paramètres à estimer dans le calcul du R^2

$$R^2_{adj} = 1 - \frac{N-1}{N-p-1}(1-R^2) = 1 - \frac{SSE/(N-p-1)}{SST/(N-1)}.$$

Statistique de Fisher La statistique de Fisher, dont l'utilisation est justifiée dans le cas explicatif de criblage car basée sur une qualité d'ajustement, est aussi utilisée à titre indicatif pour comparer des séquences de modèles emboîtés. Notons respectivement SSR_q, SSE_q et R^2_q les sommes de carrés et le coefficient de détermination du modèle réduit à $(p-q)$ termes. La statistique partielle de Fisher est

$$F = \frac{(SSR - SSR_q)/q}{SSE/(N-p-1)} = \frac{R^2 - R^2_q}{1-R^2} \times \frac{N-p-1}{q}.$$

Si l'accroissement $(R^2 - R^2_q)$ est suffisamment grand

$$R^2 - R^2_q > \frac{q}{N-p-1} F_{\alpha;q;N-p-1},$$

alors l'ajout des q termes est justifié. La notation $F_{\alpha;q,N-p-1}$ désigne le quantile d'ordre α d'une loi de Fisher à q et $N-p-1$ degrés de liberté.

Cp de Mallows L'indicateur du Cp de Mallows est une estimation de l'erreur quadratique moyenne de prévision. Le modèle complet comporte p termes. Son erreur quadratique moyenne est $MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$. Si $(p-q)$ termes sont sélectionnés parmi les p, on notera SSE_q la somme des carrés résiduels pour le modèle à $(p-q)$ termes. L'indicateur s'écrit de la manière suivante

$$Cp = \frac{SSE_q}{S^2} - N + 2(p-q),$$

où S^2 est la moyenne quadratique des résidus après régression sur les p termes. Elle peut être estimée par MSE. Il est d'usage de rechercher un modèle qui minimise le Cp, c'est-à-dire un modèle réduit biaisé mais d'estimation plus précise que le modèle complet.

Critère d'Akaike On trouve également le critère d'information d'Akaike (AIC). Ce critère s'écrit

$$AIC = 2p - 2\ln L,$$

où p est le nombre de termes du modèle et L est la valeur maximale de la fonction de vraisemblance du modèle. Parmi plusieurs choix de modèles, on retiendra celui qui a la plus faible valeur d'AIC. Ce critère repose sur un compromis entre la qualité de l'ajustement et la complexité du modèle, en pénalisant les modèles ayant un grand nombre de paramètres, ce qui limite les effets de sur-ajustement.

Critère bayésien Dans le même style, on trouve le critère d'information bayésien (BIC), définit comme

$$BIC = -2\ln L + p\ln N,$$

avec N le nombre d'observations dans l'échantillon d'apprentissage et p le nombre de termes dans le modèle. L'AIC pénalise le nombre de paramètres moins fortement que le BIC.

PRESS de Allen Enfin, on dispose de l'indice $PRESS$ de Allen, ancêtre de la validation croisée leave-one-out. On désigne par $\hat{y}_{(i)}$ la prévision de y_i calculée sans tenir compte de la $i$ème observation. La somme des erreurs quadratiques de prévision ($PRESS$) est définie par

$$PRESS = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_{(i)}\right)^2.$$

Cet indicateur permet de comparer les capacités prédictives de deux modèles.

3.2.2 Algorithme de sélection de variables

La sélection de variables par la sélection de modèles paramétriques peut se faire par différents algorithmes basé sur des stratégies différentes :

- la sélection pas à pas,
- la recherche exhaustive,
- la sélection par pénalisation.

Sélection de variables pas à pas La recherche de modèle peut se faire suivant 3 méthodes pas à pas :

- Sélection (forward) : on démarre du modèle nul et à chaque étape, un terme du modèle

maximal est ajouté au modèle en cours, c'est le terme le plus significatif. La procédure s'arrête lorsque tous les termes sont introduits ou qu'un critère d'arrêt est atteint.

- Élimination (backward) : on démarre du modèle complet et à chaque étape, le terme le moins significatif est retiré du modèle. La procédure s'arrête lorsque les variables restant dans le modèle satisfont un critère d'arrêt.
- Mixte : cet algorithme introduit une étape d'élimination après chaque étape de sélection afin de retirer du modèle d'éventuels termes devenus moins significatifs du fait de la présence de ceux nouvellement introduits.

Pour chacune des 3 méthodes décrites, le critère AIC ou BIC peut être évalué à chaque étape lors d'un ajout ou d'une suppression d'un terme du modèle. Le choix du terme se fait selon la minimisation d'un des 2 critères.

On peut aussi utiliser la statistique de Fisher. Ceci consiste donc à effectuer une analyse de la variance (ANOVA) à chaque étape de recherche du modèle. L'ANOVA (Miller, 1997) est un test statistique qui consiste à vérifier que plusieurs échantillons sont issus d'une même population. Le but est ici de comparer un modèle à ce même modèle auquel on a ajouté ou retiré un terme afin de décider si ce terme est indispensable ou non au modèle. Pour la méthode forward, le terme ajouté est celui correspondant à une p-value minimale. Cette p-value (probabilité d'obtenir la même valeur du F de Fisher si les 2 modèles sont identiques) est associée à la statistique partielle du test de Fisher qui compare les 2 modèles. Le critère d'arrêt est le fait que la p-value reste plus grande qu'une valeur seuil fixée (par défaut 0.5). Pour la méthode backward, les termes avec les plus grandes p-value sont retirés à chaque itération. Le critère d'arrêt est le fait que les termes restants ont des p-values plus petites qu'un seuil fixé (par défaut 0.1).

Recherche exhaustive Cette méthode (Furnival and Wilson, 1974) est un algorithme de sélection de variables global. Le but est de comparer tous les modèles possibles en cherchant à optimiser un des critères : R^2, R^2 ajusté ou Cp de Mallows, choisi par l'utilisateur. Tous les modèles sont considérés. La méthode propose les k meilleurs modèles pour chaque niveau, c'est-à-dire pour chaque nombre de termes présents dans le modèle. On obtiendra donc k modèles à 1 terme, k à 2 termes et ainsi de suite jusqu'à d termes, d étant le nombre total de facteurs. Le nombre k est choisi par l'utilisateur.

En général, le R^2 tend à prendre tous les termes tandis que les deux autres critères conduisent réellement à une sélection.

Sélection de modèle par pénalisation La méthode de pénalisation Ridge (Grandvalet, 1998) permet de trouver un modèle linéaire par une régression Ridge. On se place dans le

modèle linéaire

$$y = \tilde{X}\tilde{\beta} + \epsilon,$$

où

$$\tilde{X} = \begin{pmatrix} 1 & x_1^1 & x_2^1 & \dots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \dots & x_d^2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_1^N & x_2^N & \dots & x_d^N \end{pmatrix}, \tilde{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_d \end{pmatrix}.$$

On note $x^0 = (1, 1, \dots, 1)'$, X la matrice $\tilde{X}$ privée de sa première colonne et β le vecteur $\tilde{\beta}$ privé de son premier élément. L'estimateur Ridge est défini par un critère des moindres carrés avec une pénalité de type $\mathcal{L}^2$:

$$\hat{\beta}_{Ridge} = \underset{\beta \in \mathbb{R}^{d+1}}{\arg\min} \left(\sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{d} x_j^i \beta_j \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2 \right) = (X'X + \lambda I_d)^{-1} X'Y,$$

où λ est un paramètre positif à choisir. Plus la pénalité λ augmente et plus la solution obtenue est régulière ou encore, plus le biais augmente et la variance diminue. Il y a sur-ajustement avec une pénalité nulle (le modèle passe par tous les points mais oscille dangereusement) et il y a sous-ajustement avec une pénalité trop grande.
La régression Ridge permet de contourner les problèmes de colinéarité même en présence d'un nombre important de variables d'entrées. La principale faiblesse de cette méthode est liée aux difficultés d'interprétation car, sans sélection, toutes les variables sont concernées dans le modèle. D'autres approches par régularisation permettent également une sélection, c'est le cas de la régression LASSO.

La méthode Lasso (Tibshirani, 1996) correspond à la minimisation d'un critère des moindres carrés avec une pénalité de type $\mathcal{L}^1$ (et non plus $\mathcal{L}^2$ comme dans la régression Ridge). On note $||\beta|| = \sum_{j=1}^{d} |\beta_j|$. L'estimateur Lasso de β dans le modèle

$$y = X\beta + \epsilon$$

est défini par

$$\hat{\beta}_{Lasso} = \underset{\beta \in \mathbb{R}^{d}}{\arg\min} \left(\sum_{i=1}^{N} \left(y_i - \sum_{j=0}^{d} x_j^i \beta_j \right)^2 + \lambda \sum_{j=1}^{d} |\beta_j| \right),$$

où λ est un paramètre positif à choisir.
Comme dans le cas de la régression Ridge, le paramètre λ est un paramètre de régularisation. Si $\lambda = 0$, on retrouve l'estimateur des moindres carrés. Si λ tend vers l'infini, on annule tous les estimateurs des coefficients. La solution obtenue est dite parcimonieuse car elle comporte beaucoup de coefficients nuls.

Une autre pénalisation consiste à combiner la régression Ridge et la régression Lasso en introduisant les deux types de pénalités simultanément. Il s'agit de la méthode Elastic Net pour laquelle le critère à minimiser est

$$\sum_{i=1}^{N}(y_i - \beta_0 - \beta_1 x_1^i - \ldots - \beta_d x_d^i)^2 + \lambda \left(\alpha \sum_{j=1}^{d} |\beta_j| + (1-\alpha) \sum_{j=1}^{d} \beta_j^2 \right).$$

Pour $\alpha = 1$, on retrouve la méthode Lasso. Pour $\alpha = 0$, on retrouve la régression Ridge.

3.2.3 Autres méthodes de sélection de modèles

Les méthodes de régression sur composantes orthogonales permettent de modéliser la réponse quand le nombre d'entrées est important et que certaines d'entre elles sont fortement corrélées voire colinéaires. L'idée de ces méthodes est de construire des variables latentes, combinaisons linéaires des entrées. Cette construction peut se faire de deux façons, ce qui conduit à deux méthodes : méthode PLSR (Partial Least Squares Regression) et PCR (Principal Component Regression).
La régression PLS (Tenenhaus, 1998) s'inspire de l'analyse en composantes principales (ACP) et de la régression. Dans cette méthode, les entrées sont regroupées par blocs, chaque bloc étant représenté par une variable latente, appelée aussi composante. Puis une régression simple de la sortie sur ces variables latentes est effectuée. L'idée de la méthode est de commencer par une seule composante et de réaliser la régression. Si le pouvoir explicatif de cette régression est trop faible, une deuxième composante est construite et on effectue une deuxième régression sur les deux composantes. Le nombre de composantes à retenir est généralement choisi par validation croisée à l'aide des critère SSE et $PRESS$.
La régression sur composantes principales (Xie and Kalivas, 1997) permet d'obtenir un modèle de régression linéaire sur les composantes principales de l'ACP menée sur les entrées. La particularité est que l'ACP est menée, ce qui donne les composantes principales, puis la régression est effectuée par moindres carrés ordinaires sur les composantes, sans les modifier. Enfin, les coefficients de la régression subissent une transformation pour être exprimés selon l'échelle des

entrées initiales afin de caractériser le modèle d'origine.

Une autre méthode est la méthode BMA (Bayesian Model Averaging). Cette méthode bayésienne (Hoeting et al., 1998) consiste à prendre en compte l'incertitude sur la forme du modèle ou sur les hypothèses de ce modèle et de la propager jusqu'aux conséquences sur une quantité d'intérêt en sortie.

Les méthodes de sélection de modèles paramétriques sont nombreuses et bien adaptées lorsque l'objectif est d'obtenir un méta-modèle pour une relation assez régulière entre la sortie et les entrées du système. Dans le cas de fortes non-linéarités, il existe plusieurs méta-modèles basés sur une régression non-paramétrique qui permettent d'effectuer une sélection de variables. C'est notamment le cas de la projection par directions révélatrices, la régression par partitionnement récursif, les modèles additifs et la méthode MARS expliquées en Annexe B. La méthode LOESS, expliquée à la Section 4, permet également de réaliser une sélection de variables. La méthode, expliquée par (Storlie and Helton, 2008) pour tous les méta-modèles sauf MARS, consiste à effectuer une sélection pas à pas par ajout successif de termes au modèle.
Il est également possible de construire le méta-modèle sur toutes les entrées et d'effectuer une sélection de variables a posteriori. Il s'agit de l'analyse de sensibilité.

3.3 Sélection par analyse de sensibilité

L'analyse de sensibilité repose souvent sur la décomposition de la variance fonctionnelle. Cette décomposition consiste d'abord à exprimer la fonction f en une somme de fonctions élémentaires

$$f(x_1,\dots,x_d) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{i=1}^{d-1}\sum_{j=2,j>i}^{d} f_{ij}(x_i,x_j) + \dots + f_{12\dots d}(x_1,\dots,x_d), \tag{II.11}$$

où f doit être intégrable sur le domaine d'entrée D normalisé de telle sorte que $D = [0,1]^d$. La fonction f_0 est une constante, les autres vérifient pour tout $k = 1,\dots,s$ et pour tout $\{i-1,\dots,i_s\} \subset \{1,\dots,d\}$

$$\int_{-1}^{1} f_{i_1,\dots,i_s}(x_{i_1},\dots,x_{i_s})dx_{i_k} = 0. \tag{II.12}$$

Les fonctions f_i sont des fonctions d'une variable représentant les effets principaux, les fonctions f_{ij} sont des fonctions de deux variables représentant les interactions d'ordre 2, etc. Introduite

par (Sobol', 1993) pour l'analyse de sensibilité, cette décomposition est unique et tous les termes de II.11 peuvent être évalués de la manière suivante

$$f_0 = \mathbb{E}(Y), \tag{II.13}$$
$$f_i(x_i) = \mathbb{E}(Y|X_i) - \mathbb{E}(Y), \tag{II.14}$$
$$f_{ij}(X_i, X_j) = \mathbb{E}(Y|X_i, X_j) - f_i - f_j - \mathbb{E}(Y), \tag{II.15}$$

où la sortie et les entrées sont considérés comme des variables aléatoires avec $\mathbb{E}(Y)$ et $\mathbb{E}(Y|X_i)$ respectivement l'espérance et l'espérance conditionnelle de la sortie Y. Pour les termes d'ordres plus grands, on obtient le même type de formules.
Si les X_i sont indépendantes, alors l'Equation (II.11), avec les relations (II.13), (II.14) et (II.15), conduit à la décomposition de la variance fonctionnelle (FANOVA)

$$Var(Y) = \sum_{i=1}^{d} V_i(Y) + \sum_{i=1}^{d-1} \sum_{j=2, j>i}^{d} V_{ij}(Y) + \ldots + V_{12\ldots d}(Y), \tag{II.16}$$

où $V_i(Y) = Var(\mathbb{E}(Y|X_i)), V_{ij}(Y) = Var(\mathbb{E}(Y|X_i, X_j)) - V_i(Y) - V_j(Y)$, etc. Les indices de sensibilité sont obtenus à partir de (II.16)

$$S_i = \frac{V_i(Y)}{Var(Y)}, S_{ij} = \frac{V_{ij}(Y)}{Var(Y)}, \ldots, S_{12\ldots d} = \frac{V_{12\ldots d}(Y)}{Var(Y)}. \tag{II.17}$$

Appelées aussi indices de Sobol, ces quantités sont comprises entre 0 et 1 et sont de somme égale à 1. Chaque indice représente la part de variance de la sortie expliquée par une entrée ou une interaction. Ainsi, les indices du premier ordre S_i désignent la sensibilité de la sortie à l'entrée i, les indices du deuxième ordre S_{ij} expriment la sensibilité de la sortie à l'interaction entre X_i et X_j, et ainsi de suite.
Pour un problème à d variables, on obtiendra $2^d - 1$ indices. Ce nombre peut donc devenir rapidement problématique, d'où l'introduction dans (Homma and Saltelli, 1996) des indices totaux, qui représentent l'effet total d'une variable sur la sortie (effet principal plus toutes les interactions)

$$S_{T_i} = S_i + \sum_{j=1, j\neq i}^{d} S_{ij} + \ldots S_{12\ldots d}. \tag{II.18}$$

Pour un grand nombre d'entrée d, on pourra se contenter des indices du premier ordre et des indices totaux.

Il existe plusieurs méthodes pour estimer ces indices de sensibilité :

- échantillons de Monte-Carlo (ou méthode de Sobol),
- échantillons de quasi Monte-Carlo,
- la méthode FAST,
- la méthode de McKay,
- l'approche non-paramétrique des modèles additifs,
- l'utilisation d'un méta-modèle de f.

Les échantillons obtenus par simulations de Monte-Carlo (Saltelli, 2002; Sobol', 1993) ou de quasi Monte-Carlo (Saltelli et al., 2008) permettent d'estimer empiriquement les différentes variances. Ils doivent donc être suffisamment grands pour obtenir des estimations précises, ce qui rend ces deux méthodes coûteuses, surtout la première. La méthode FAST (Cukier et al., 1978) est basée sur une transformée de Fourier multidimensionnelle de f. Cette méthode, moins coûteuse que les précédentes, permet d'évaluer les indices totaux mais reste toujours coûteuse et ne supporte pas très bien l'augmentation du nombre d'entrées. La méthode de McKay (McKay et al., 1995) est basée sur l'échantillonnage par hypercubes latins répliqués. Les modèles additifs décrits en Annexe B permettent de calculer des indices d'ordre 1 équivalents aux indices de Sobol. Le problème de cette méthode est que le mauvais ajustement de f par le modèle additif conduira à des erreurs importantes d'estimation des indices. L'utilisation du méta-modèle permet de réduire les temps de calculs en effectuant des simulations de Monte-Carlo dont les points sont évalués avec le méta-modèle et plus directement avec le code. Il est nécessaire d'avoir un méta-modèle qui représente bien f pour obtenir une bonne estimation des indices.

Selon l'expérience qui en a été faite dans cette thèse, les indices de Sobol donnent des résultats plus précis que le criblage mais sont assez équivalents à ceux obtenus avec la sélection de modèles. Le criblage est la méthode la plus économe en termes de nombre d'appels à la fonction, elle est d'ailleurs le plus souvent utilisée pour les campagnes d'essais physiques. Les deux autres méthodes nécessitent de pouvoir faire suffisamment de calculs pour établir un modèle. Ils demandent donc plus d'appels à la fonction que le criblage mais fournissent souvent des résultats plus précis.
Il existe d'autres méthodes de réduction de la dimension, basées sur la recherche d'un méta-modèle. Certaines méthodes de méta-modélisation sont décrites dans la section suivante.

4 Méthodes de méta-modélisation

Les codes de calcul complexes sont utilisés pour simuler, par exemple, un phénomène physique et sont considérés par l'utilisateur comme étant le modèle de référence du système étudié. Il n'est pas toujours facile d'utiliser directement le code de calculs pour traiter les problèmes usuels d'incertitudes, de sensibilité, d'optimisation ou encore de robustesse. D'autant que leur complexité peut rendre les temps de calcul très élevés. La méthode permettant de pallier ces problèmes consiste à remplacer le code de calcul coûteux par une fonction mathématique peu coûteuse à évaluer et appelée méta-modèle. La démarche générale est de considérer que le modèle de référence est une boîte noire, c'est-à-dire un système dont le fonctionnement interne est soit inaccessible, soit délibérément masqué. Mais certains méta-modèles peuvent être construits en tenant compte de la connaissance du problème initial, ce qui est le cas lorsqu'on dispose d'une boîte blanche par exemple.
De nombreux termes équivalents à "méta-modèle" peuvent être trouvés dans la littérature comme :

- modèle de substitution,
- surface de réponse,
- proxy,
- surrogate...

Dans cette section, nous proposons de passer en revue différents types de méta-modèles : leur utilité, leur construction, leurs avantages et leurs inconvénients. Nous y voyons également les différents critères existants afin de valider le choix d'un méta-modèle. Mais d'abord, voyons plus en détail le contexte de l'utilisation de ces techniques de méta-modélisation.

4.1 Contexte d'utilisation des méta-modèles

Avec l'augmentation des puissances de calcul et l'amélioration des outils de simulation, la précision des modèles numériques est en pleine croissance, ce qui permet de se rapprocher de plus en plus des phénomènes physiques réels. Cette amélioration conduit cependant à une augmentation des temps de calculs pouvant devenir rédhibitoires dans les études exigeant un grand nombre d'appels au code. Le principe de la méta-modélisation, représenté à la Figure II.8, est de substituer le modèle physique de référence par un modèle approché, appelé méta-modèle, pour réduire le coût de calcul.
Le méta-modèle considère les mêmes entrées et les mêmes sorties que le modèle physique mais en fournit une approximation simplifiée, plus manipulable et moins coûteuse en temps de cal-

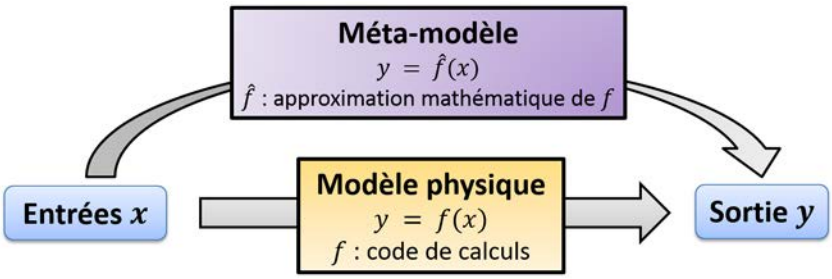

Figure II.8 – Schéma de la méta-modélisation

culs. En général, un méta-modèle est considéré pour chaque sortie du système. Les entrées peuvent être très nombreuses et de différents types (continues, discrètes, binaires, etc).

Nous avons défini la méta-modélisation comme la méthode pour approximer un modèle physique. Mais elle peut aussi être utilisée dans le but d'explorer le domaine de conception. Ceci permet à l'utilisateur de mieux comprendre les problèmes de conception en travaillant avec un méta-modèle peu coûteux. La méta-modélisation est aussi utilisée dans la formulation de problème. Ceci permet de réduire le nombre de variables de conception ou leur domaine de variation. Par cela, il est également possible d'éliminer des contraintes inutiles ou de reformuler un problème afin qu'il soit plus simple à résoudre ou plus précis.
Ces applications constituent une base commune très utile pour des applications majeures telles que

- la propagation d'incertitudes : certaines entrées sont incertaines et leur variabilité se répercute sur les sorties du système ;
- l'analyse de sensibilité : les sorties du système sont plus ou moins sensibles aux différentes entrées dont les impacts sur la variabilité d'une sortie peuvent être quantifiés et hiérarchisés ;
- l'inversion : on cherche les entrées permettant d'atteindre une certaine valeur de la sortie ;
- l'optimisation : on cherche les entrées permettant de minimiser ou de maximiser une ou plusieurs sorties ;
- la conception robuste : on cherche à optimiser une sortie tout en prenant en compte les incertitudes dont le système est entaché afin que l'optimum y soit résistant.

Ces applications sont représentées à la Figure 2 (Wang and Shan, 2007) par deux demi-ellipses : celle du bas regroupe les applications de base servant aux applications majeures regroupées dans la demi-ellipse du haut. Maintenant que nous avons exposé le contexte d'utilisation des méta-modèles, voyons quel en est le principe.

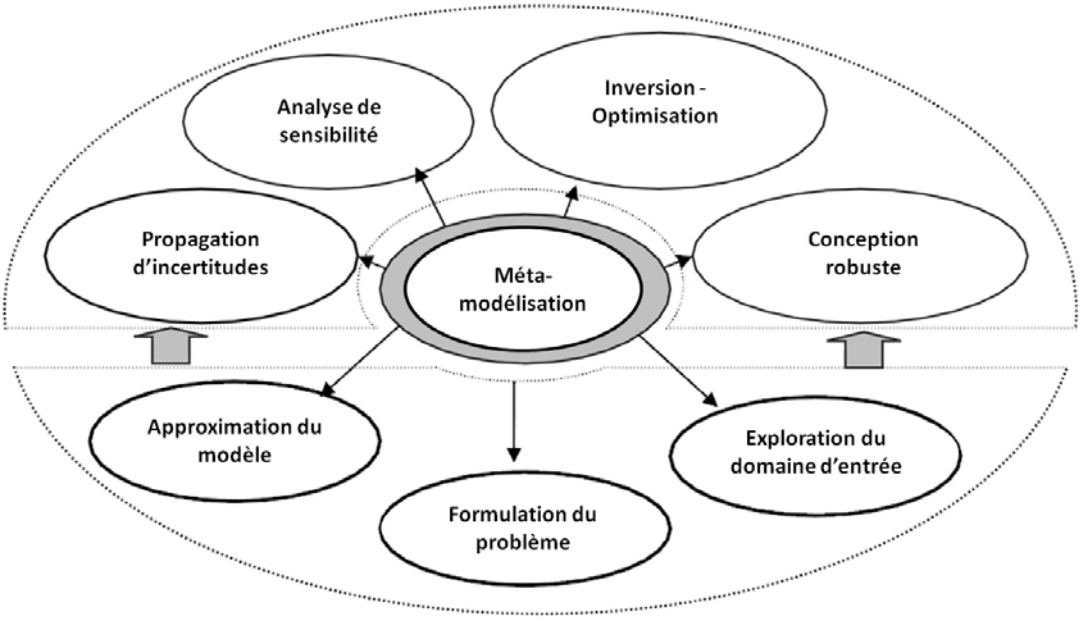

Figure II.9 – Utilisation de la méta-modélisation dans différents problèmes de conception

4.2 Principe de la méta-modélisation

Afin de faciliter l'exploitation du système, nous cherchons à construire un modèle mathématique en utilisant les observations de couples entrées-sorties issues des modèles de simulation existants mais aussi les connaissances que nous pouvons avoir sur ce système. La démarche générale de la méta-modélisation peut être présentée en trois étapes :

- L'*échantillonnage* consiste à effectuer quelques simulations du modèle de référence. Il s'agit de réaliser un plan d'expériences numérique constituant un échantillon d'apprentissage.
- La *construction* du méta-modèle utilise l'échantillon pour exprimer explicitement le méta-modèle. Cette étape fait appel à des méthodes d'ajustement du méta-modèle. Elle inclut également l'évaluation de l'erreur d'approximation qui est fondamentale puisqu'elle doit valider les résultats fournis par le méta-modèle ou, au minimum, donner un indice de confiance.
- L'*exploitation* est la phase d'utilisation du méta-modèle, elle permet d'extraire les résultats souhaités pour un ou plusieurs des problèmes proposés à la Figure II.9. Pour la demi-ellipse du haut, une méthode de type Monte-Carlo (ou une autre méthode de simulation) peut être employée étant donné que le coût d'une simulation sur le méta-modèle est très faible.

Dans ce chapitre, nous nous intéresserons principalement aux deux premières étapes de méta-modélisation. La dernière étape sera effectuée dans les deux chapitres suivants, dans lesquels les méta-modèles seront utilisés.

Il existe de nombreux types de méta-modèles que l'on peut classer de différentes façons :

- intrusifs (conséquences sur le code) ou non,
- paramétriques, non-paramétriques ou semi-paramétriques,
- linéaires ou non,
- interpolants (ajustement exact des points de l'échantillon d'apprentissage) ou non,
- issus de la classification (apprentissage supervisé) ou non.

Pour chaque type de méta-modèle, on choisit une méthode d'échantillonnage et une méthode d'ajustement pour le construire. Les méthodes d'échantillonnage les plus connues ainsi que les méta-modèles les plus utilisés et les méthodes d'ajustement possibles sont répertoriés dans le Tableau II.4. Dans le but d'être les plus exhaustifs possibles, nous avons complété les tableaux issus de (Wang and Shan, 2007) et de (Simpson et al., 2001). Dans leur article, Simpson et ses co-auteurs établissent des associations usuelles entre un type de méta-modèle et les méthodes d'échantillonnage et d'ajustement les plus adaptées. Ils distinguent notamment quatre techniques d'approximation :

1. La méthodologie de surface de réponse consiste à établir un modèle polynomial à partir d'un plan factoriel ou un plan composite centré (ces plans sont détaillés dans la Section 3). La méthode d'ajustement la plus couramment utilisée avec ce type de modèle est la régression par moindres carrés.
2. Le krigeage permet d'établir un modèle non-paramétrique basé sur la réalisation d'un processus stochastique à partir de plans D-optimaux ou de plans latins hypercubes. La méthode d'ajustement repose sur le meilleur estimateur linéaire non biaisé (*BLUE* en anglais).
3. Les réseaux neuronaux permettent d'établir un modèle par réseaux de neurones à partir de plans établis à la main, avec une méthode d'ajustement basée sur la rétro-propagation.
4. L'apprentissage inductif permet d'établir des arbres de décision avec un critère d'ajustement basé sur l'entropie.

4.2.1 Méthodes d'échantillonnage

Les plans d'expériences classiques sont le plus souvent utilisés pour les expériences physiques. Parmi les méthodes classiques, les plans factoriels sont décrits dans la Section 3, puisqu'ils sont

Méthodes d'échantillonnage	Choix de méta-modèle	Méthodes d'ajustement

- Méthodes classiques
  • Factoriel (fractionnaire)
  • Central composite
  • Box-Behnken
  • Optimal alphabétique
  • Plackett-Burman
- Méthodes "space-filling"
  • Grilles simples
  • Hypercube latin
  • Tableaux orthogonaux
  • Suite de Hammersley
  • Plans uniformes
  • Minimax et maximin
- Méthodes hybrides
- Sélection aléatoire ou par l'utilisateur
- Importance sampling
- Simulation directionnelle
- Échantillonnage discriminant
- Méthodes séquentielles ou adaptatives
	- Polynomial (linéaire, quadratique ou de degré supérieur)
	- Splines (linéaire, cubiques, NURBS)
	- Modèles additifs généralisé (GAM)
	- Régression localement polynomiale (LOESS)
	- Splines de régression multivariées adaptatives (MARS)
	- Régression par directions révélatrices (PPR)
	- Lissage à noyau
	- Méthodes COSSO, ACOSSO
	- Réalisations d'un processus stochastique (krigeage)
	- Fonctions à base radiale (RBF)
	- Polynômes d'interpolation de plus petit degré (Lagrange)
	- Réseaux de neurones
	- Base de règles ou arbre de décision (partitionnement récursif)
	- Forêts aléatoires (RF)
	- Décomposition sur base connue (chaos polynomial, Fourier, ondelettes...)
	- Machine à support de vecteur (SVM)
	- Modèles hybrides
		- Régression par moindres carrés (pondérés)
		- Meilleur prédicteur linéaire non biaisé
		- Meilleur prédicteur linéaire
		- Log-vraisemblance
		- Approximation multipoint (MPA)
		- Méta-modélisation séquentielle ou adaptative
		- Rétro-propagation
		- Entropie

Tableau II.4 – Techniques de méta-modélisation couramment utilisées

utiles pour la réduction de la dimension. Le lecteur intéressé par les plans centraux composites, Box-Behnken et Plackett-Burman pourra se référer à (Myers and Montgomery, 2009). Les plans optimaux alphabétiques (on parle de critères d'A-optimalité, C-optimalité, D-optimalité, etc), sont décrits dans le chapitre 6 de (Pukelsheim, 1993).
Dans tous ces plans, les points de l'échantillon sont répartis sur les bornes de l'espace de conception, plus quelques expériences au centre du domaine. Le but est de minimiser l'influence de l'erreur aléatoire des expériences physiques sur le problème considéré. Or, comme ceci est expliqué dans (Wang and Shan, 2007), les expériences sur ordinateur impliquent davantage une erreur systématique qu'une erreur aléatoire. Dans ce cas-là, il est préférable de choisir un plan d'expérience qui quadrille bien le domaine (plans « space-filling ») plutôt que de concentrer les expériences sur les bornes du domaine. Les plans classiques s'avèrent également inefficaces et même inappropriés pour les codes de calculs déterministes (plusieurs lancements d'une même expérience donneront toujours le même résultat). Comme cela est décrit dans (Simpson et al., 2001), on préfèrera donc également les plans « space-filling » dans le cas des codes de calculs déterministes.

Parmi les méthodes « space-filling », les plus utilisées sont les tableaux orthogonaux (Hedayat et al., 2012), les plans latins hypercubes (McKay et al., 1979), les suites d'Hammersley (Kalagnanam and Diwekar, 1997) et les plans uniformes (Fang et al., 2000). Ces quatre méthodes sont comparées dans (Simpson et al., 2001). Les deux dernières appartiennent à un groupe plus général appelé suites à faible discrépance. La faible discrépance est un critère d'uniformité qui fait partie des trois propriétés que les plans « space-filling » doivent respecter avec le remplissage et l'indépendance.
Les plans les plus utilisés dans l'industrie sont les plans hypercubes latins (LHS), plans décrits dans la Section 3. Certains critères permettent d'optimiser les plans latin. Il s'agit par exzemple des plans minimax et maximin, qui sont décrits dans (Johnson et al., 1990).

La méthode de simulation par Monte-Carlo (MCS), méthode d'échantillonnage aléatoire moins efficace que les méthodes présentées précédemment, est encore très utilisée dans l'industrie. L'*Importance Sampling* (IS), que l'on peut traduire par échantillonnage préférentiel (Au and Beck, 1999), est une amélioration de la méthode MCS dans le sens où elle permet de réduire la variance de l'estimateur tout en maintenant le même niveau de précision que MCS. Une autre variante de MCS est la simulation directionnelle (Ditlevsen et al., 1986). L'échantillonnage discriminant (Wang et al., 2005) a quant à lui été développé spécialement pour l'optimisation.

Les méthodes séquentielles et adaptatives sont de plus en plus utilisées, principalement à cause de la difficulté de connaître a priori la bonne taille de l'échantillon d'apprentissage. Certaines de ces méthodes sont comparés dans (Jin et al., 2002).

Les nombreuses méthodes d'échantillonnage permettent d'obtenir des points d'observation du domaine d'entrée. Ces points forment ce que l'on appelle l'échantillon d'apprentissage. Cet échantillon est ensuite utilisé pour construire le méta-modèle. Comme nous l'avons vu et comme ceci est précisé dans (Simpson et al., 2001), les méthodes d'échantillonnage « space-filling » sont les plus adaptées pour les codes de calculs déterministes. Nous considérerons donc des plans latins hypercubes, plans simples à construire et implémentés dans de nombreux logiciels industriels, dans les études de méta-modélisation effectués dans la suite.

4.2.2 Choix du méta-modèle et de la méthode d'ajustement associée

Une fois le plan d'expériences approprié sélectionné et les expériences nécessaires menées, la prochaine étape consiste à choisir un méta-modèle et une méthode d'ajustement. Comme nous l'avons vu dans le Tableau II.4, de nombreux méta-modèles et de nombreuses méthodes d'ajustement existent. Dans la littérature, beaucoup d'auteurs ont proposé des comparaisons de certaines de ces méthodes appliquées à des exemples tests ou des cas réels. Dans (Villa-Vialaneix et al., 2012) est proposée une comparaison de 8 techniques de méta-modélisation appliquées à un cas environnemental d'agriculture intensive. Les auteurs comparent 2 modèles linéaires (le modèle linéaire général avec tous les termes d'ordre 1 et un modèle avec les termes linéaires, des termes non-linéaires et des interactions) avec 6 méthodes non-paramétriques (2 basées sur les splines (ACOSSO et SDR-ACOSSO), une approche par krigeage, un réseau de neurones, une méthode SVM et une méthode de forêts aléatoires). Dans (Jin et al., 2001), quatre techniques de méta-modélisation (régression polynomiale, krigeage, MARS et RBF) sont proposées sur 14 problèmes tests. Chacun de ces problèmes est différent des autres selon son taux de non-linéarité, le nombre de variables d'entrée et le comportement bruité. Les 13 premiers sont des problèmes mathématiques, le dernier est un problème industriel réel sur la tenue de route d'un véhicule. Dans (Storlie et al., 2009), des méthodes de régression non-paramétriques (MARS, RF, la régression par augmentation de gradient (GBR), qui utilise les arbres de régression, ACOSSO et le krigeage) sont proposées pour l'analyse de sensibilité sur des exemples simulés et un cas réel de confinement de déchets.

Toutes ces méthodes ne seront pas décrites ici. Celles que nous avons utilisées dans le cadre de

la thèse seront expliquées. D'autres méta-modèles connus et souvent utilisés en pratique seront expliqués en Annexe **??**, sans détails techniques.

Avant de décrire ces méthodes de méta-modélisation, rappelons quelques notations du Chapitre I. On dispose de d entrées indépendantes que l'on regroupe dans un vecteur $x = (x_1, ..., x_d)$ et d'une sortie d'intérêt y. Les code de calculs reliant les entrées à la sortie est noté f. On a donc $y = f(x)$. Le méta-modèle est une fonction $\hat{f}$ qui vérifie

$$y = \hat{f}(x) + \epsilon, \tag{II.19}$$

où ϵ représente l'erreur de modélisation. On note $\hat{y}$ l'estimation de y par le méta-modèle $\hat{f}$: $\hat{y} = \hat{f}(x)$.

Régression paramétrique Les modèles de régression paramétrique sont populaires grâce à leur simplicité et au fait que si les sorties du système sont approximativement linéaires vis-à-vis des entrées, ces méthodes sont bien adaptées. De plus, ces méthodes permettent d'obtenir une expression analytique du modèle puisqu'elles sont paramétriques. Ceci sous-entend donc que la forme du modèle est choisie a priori. En régression paramétrique, on peut choisir parmi cinq types de modèles :

- polynomial,
- sur les rangs,
- linéaire généralisé,
- non linéaire,
- réseaux de neurones.

De nombreux auteurs se sont intéressés à ces méthodes, notamment, par ordre chronologique : (Barton, 1992), (Barton, 1994), (Wang et al., 1999), (Jin et al., 2001), (Simpson et al., 2001), (Storlie and Helton, 2008), (Storlie et al., 2009) et(Villa-Vialaneix et al., 2012).
Nous allons nous intéresser aux quatre premiers types de modèles. Les réseaux de neurones sont expliqués en Annexe B.

Régression polynomiale Ces méta-modèles sont les surfaces de réponse les plus utilisées en pratique. Le but est de considérer le modèle comme étant un polynôme de degré minimal. Pour les faibles courbures, un polynôme d'ordre 1 peut être utilisé, comme dans l'équation (II.20). On parle alors de régression linéaire. Pour une courbure significative, il est possible de considérer un polynôme d'ordre 2 avec toutes les interactions d'ordre 2, comme dans l'équation (II.21).

On parle alors de régression quadratique.

$$\hat{y} = \beta_0 + \sum_{i=1}^{d} \beta_i x_i \tag{II.20}$$

$$\hat{y} = \beta_0 + \sum_{i=1}^{d} \beta_i x_i + \sum_{i=1}^{d} \beta_{ii} x_i^2 + \sum_{i=1}^{d} \sum_{j=1, i<j}^{d} \beta_{ij} x_i x_j \tag{II.21}$$

Les paramètres des polynômes sont déterminés par moindres carrés en supposant que les erreurs ϵ sont des variables aléatoires indépendantes et identiquement distribuées selon une loi normale centrée et de variance connue σ^2. Ceci implique que $Y = \hat{f}(X) + \epsilon$ suit une loi normale $\mathcal{N}(\hat{f}(X), \sigma^2)$.

Ce type de méta-modèles présente de nombreux avantages. En effet, le cas d'un vecteur d'entrée de grande taille est facilement pris en compte. De plus, la formulation analytique obtenue est simple, facilement exploitable et compréhensible, ce qui rend la prédiction de nouveaux points très rapide. De telles méthodes sont d'ailleurs souvent utilisées pour effectuer de la réduction de dimension par screening (cf Section 3).
En revanche, ces méta-modèles exigent souvent un grand nombre d'expériences pour construire le modèle. En effet, plus l'échantillon d'apprentissage est grand, plus les connaissances sur le code sont grandes, plus le modèle sera bon. En général, pour d variables d'entrée, on considèrera au moins $10d$ expériences. De plus, elles ne sont pas adaptées pour faire de l'extrapolation et n'estiment pas parfaitement le modèle aux points du plan, contrairement aux méthodes d'interpolation. On parle alors d'erreur d'apprentissage. En contrepartie, ces méthodes fournissent un bon ajustement entre les points du plan. Ainsi, si l'on considère des nouveaux points, non utilisés pour la construction du modèle, leur estimation sera bonne. De tels points forment ce que l'on appelle un échantillon de validation (ou de test). On dira que ces méthodes ont de faibles erreurs de test.
Enfin, les surfaces de réponses sont en général des polynômes du second ordre. Elles sont donc rapidement limitées pour modéliser des fonctions non-linéaires. Dans ce cas, on peut utiliser des surfaces de réponse d'ordres plus grands. Cependant, des instabilités peuvent survenir et il peut devenir difficile d'estimer tous les coefficients sans un plan important, particulièrement en grandes dimensions. Il existe alors les modèles de régression linéaire généralisée (GLM) et de régression sur les rangs.

Régression linéaire généralisée Dans ce cas, le modèle linéaire est relié à la variable réponse par une fonction lien g. On a alors

$$g(\hat{y}) = \beta_0 + \sum_{i=1}^{d} \beta_i x_i \tag{II.22}$$

La sortie y n'est alors plus forcément issue d'une loi normale mais d'une autre loi de la famille exponentielle.
Plusieurs fonctions liens existent dans la littérature comme :

- La fonction identité : on retombe sur le modèle linéaire avec une loi normale sur y.
- La fonction inverse : $g(\hat{y}) = -1/\hat{y}$. La loi de y devient une exponentielle ou une gamma.
- La fonction inverse quadratique : $g(\hat{y}) = -1/\hat{y}^2$. La loi est alors une gaussienne inverse.
- La fonction logarithme : $g(\hat{y}) = \log \hat{y}$, utilisée lorsque la sortie est une variable de comptage, qui suit alors une loi de Poisson.
- La fonction logit : $g(\hat{y}) = \log(\hat{y}/(1-\hat{y}))$, utilisée dans le cas où la sortie est discrète. Elle suit alors une loi de Bernoulli ou une binomiale.

Comme la loi n'est plus forcément normale, alors les paramètres du modèle sont estimés par la méthode du maximum de vraisemblance, généralisation de la méthode des moindres carrés.
Le problème de la régression linaire généralisée est que le choix de la fonction lien est subjectif, donc difficilement automatisable. Une autre transformation consiste à travailler sur les rangs.

Régression sur les rangs Une condition pour utiliser la transformation des rangs est la monotonie de la relation entre la sortie et les entrées (Iman and Conover, 1979). Le principe de cette transformation est simple. On considère que l'on dispose d'un échantillon d'apprentissage de N points. Pour chaque variable, on ordonne les N valeurs en attribuant le rang 1 à la première, 2 à la seconde et ainsi de suite jusqu'à la dernière qui est de rang N. En remplaçant ainsi les valeurs des variables par leur rang, on convertit une relation monotone en une relation linéaire. Une régression linéaire peut alors être effectuée sur ces rangs. Cette méthode a d'ailleurs été utilisée sur beaucoup d'analyses de sensibilité (Iman and Helton, 1991). Cependant, elle n'améliore pas la qualité d'une étude de régression lorsque les relations ne sont pas monotone ni même linéaire. Dans ce dernier cas, il est préférable d'utiliser une régression non-linéaire.

Régression non-linéaire Selon les connaissances que nous pouvons avoir sur le code, il est également possible de réaliser la régression, non plus sur les polynômes $(1, x, x^2, \ldots)$ mais sur un ensemble de fonctions contenant les polynômes et des fonctions comme l'exponentielle, le

logarithme, l'inverse, le sinus, etc. Le modèle supposé pourrait donc ressembler à l'équation (II.23). La non-linéarité peut être considérée uniquement sur quelques entrées, avec des choix de fonctions différents.

$$\hat{y} = \beta_0 + \sum_{i=1}^{d} \beta_i x_i + \sum_{i=1}^{d} \beta_{d+i} e^{x_i} \tag{II.23}$$

Les paramètres $\beta_j, j = 1...2d$ sont estimés par moindres carrés.
Le principal inconvénient de cette méthode est assez évident, il faut décider a priori de la forme que le modèle non-linéaire va prendre. De plus, l'ajustement du modèle et l'interprétation des résultats (pour l'analyse de sensibilité par exemple) sont plus difficiles à effectuer que dans le cas de la régression polynomiale.
Lorsque l'utilisateur n'a aucun a priori sur les relations entre la sortie et les entrées mais qu'il est clair qu'elles présentent des non-linéarités importantes, il est préférable d'utiliser des méthodes de régression non-paramétrique. Leur principal avantage vis-à-vis de la régression non-linéaire est qu'elles prennent en compte les relations non-linéaires sans supposer a priori une forme pour le modèle.

Régression non-paramétrique Les modèles de régression non-paramétrique sont plus flexibles que les méthodes paramétriques et permettent donc de traiter plus aisément les non-linéarités.
La régression non paramétrique, appelée aussi lissage, est basée sur l'hypothèse d'une relation entre les entrées et une sortie du système, qui serait de la forme (II.19). Cependant, f ne prend plus une forme paramétrique particulière. Certaines hypothèses peuvent parfois être prises sur f ou ses dérivées, comme des restrictions de continuité par exemple.

Les développements des méthodes non-paramétriques sont très nombreuses, elles sont énumérées dans le Tableau II.4.
Dans cette partie, nous ne développerons que les méthodes de lissage par noyaux et de régression localement polynomiale (Storlie and Helton, 2008), les autres seront expliquées en Annexe B.

Lisseurs à noyau Les lisseurs à noyau sont des méthodes de lissage basées sur des moyennes pondérées où les poids sont définis par une fonction noyau K. Ce sont des méthodes locales qui permettent de construire, à partir de N couples d'observations $(x^1, y_1), ..., (x^N, y_N)$, des

méta-modèles de la forme

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} K((x^i - x)/h) y_i}{\sum_{i=1}^{N} K((x^i - x)/h)}, \tag{II.24}$$

où h est un paramètre de lissage correspondant à la taille du voisinage considéré autour du point en cours x. Plus le paramètre h est grand, plus les données seront lissées, inversement plus il sera petit, plus la fidélité aux données sera importante. Le coefficient h contrôle la convergence de l'estimateur et l'équilibre biais-variance. Son choix a plus d'impact que celui du noyau lui-même. Le rôle de la fonction noyau est de mettre plus de poids sur les y_i associés à des x^i proches de x et moins sur ceux associés à des x^i éloignés de x. La fonction noyau doit être à valeurs positives et être d'intégrale 1. Plusieurs choix sont possibles pour la fonction noyau :

- Fenêtre glissante : $K(||x_i - x||, h) = I_{[0,d_r(x)]}(||x_i - x||)$ où $I_A(u)$ désigne la fonction indicatrice ($I_A(u) = 1$ si $u \in A$ et $I_A(u) = 0$ sinon), $d_r(x)$ est la distance maximale entre x et ses r plus proches voisins, c'est-à-dire les r valeurs x^i telles que $||x^i - x|| \leq d_r(x)$; on a alors
$$\hat{f}(x) = \frac{1}{r} \sum_{i=1}^{N} I_{[0,d_r(x)]}(||x^i - x||) y_i.$$
- Noyau de Nadaraya-Watson (lissage linéaire) : $K(z, h) = \frac{e^{(-z^2/(2h^2))}}{(2\pi)^{N/2}}$ qui correspond à la densité d'une loi normale multidimensionnelle, centrée et réduite (cf. (Nadaraya, 1964) et (Watson, 1964)).

D'autres noyaux existent comme le noyau triangulaire, celui d'Epanechnikov ou encore le noyau circulaire.

Le paramètre de lissage peut être choisi comme étant le h qui minimise l'erreur moyenne quadratique intégrée (erreur L^2)

$$MISE(h) = \mathbb{E}\left(\int (\hat{f}_h(x) - f(x))^2\right).$$

En pratique, ce coefficient est déterminé par validation croisée. L'objectif est de sélectionner le h qui minimise l'erreur empirique $\mathbb{L}^2$ modifiée

$$CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{f}_h^{(-i)}(x^i))^2,$$

où $\hat{f}_h^{(-i)}$ est l'estimateur de f qui ne tient pas compte de l'observation i.
Une autre façon d'écrire l'estimateur à noyau multivarié est de considérer une matrice de lissage H d'ordre N symétrique et définie positive, un noyau K qui est une densité multivariée

symétrique et le noyau $K_H(x) = det(H)^{-1/2}K(H^{-1/2}x)$. Le lecteur intéressé pourra se référer à (Ruppert and Wand, 1994).

Un inconvénient possible pour les estimateurs à noyau se trouve dans les effets de bord. Ceci est dû au fait que les méthodes sont basées sur des moyennes locales pondérées. En effet, ces effets peuvent se manifester au voisinage des plus petites et des plus grandes valeurs observées de x. En ces points, le nombre d'observations à droite et à gauche est en déséquilibre. Le risque est de déformer la fonction pour les valeurs de x proches des bornes du domaine.
Ces méthodes ne sont pas adaptées en grande dimension ($d > 3$). Elles sont d'ailleurs le plus souvent utilisées dans le cas d'une unique entrée. En multidimensionnel, on utilise plutôt la régression localement polynomiale.

Régression localement polynomiale Appelée LOESS (LOcally wEighted Scatterplot Smoothing) dans la littérature, la régression localement pondérée est une généralisation naturelle de l'estimateur précédent. Il permet d'établir des méta-modèles de la forme

$$\hat{f}(x) = \alpha(x) + \beta(x)x, \tag{II.25}$$

avec $\beta(x) = (\beta_1(x), ..., \beta_d(x)), x = (x_1, ..., x_d)$. Pour une valeur spécifique de x, les quantités $\alpha(x)$ et $\beta(x)$ sont définies comme étant les valeurs de α et β minimisant

$$\sum_{i=1}^{N}(\alpha + \beta x^i - y_i)^2 K(||x^i - x||, h),$$

avec K une fonction noyau définie de la manière suivante

$$K(||x^i - x||, h) = \left[1 - \left(\frac{||x - x^i||}{d_r(x)}\right)^3\right]^3 I_{]0,d_r(x)]}(||x - x^i||),$$

où $d_r(x)$ est la distance entre x et son $r$ième plus proche voisin et les d entrées sont normalisées.
La détermination de $\alpha(x)$ et $\beta(x)$ fournit un estimateur de y pour une valeur de x. Les estimateurs pour des valeurs supplémentaires de x sont obtenus en résolvant le problème de minimisation pour chaque x considéré. Cela peut alors sembler coûteux mais LOESS est une méthode rapide, même pour de grandes dimensions. L'avantage principal de cette méthode réside dans sa capacité à capter les comportements non-linéaires en grande dimension mais aussi

les interactions entre les entrées.
Un inconvénient de la méthode LOESS, comme les autres techniques par moyenne locale, est que les valeurs observées x_i les plus proches de x ne sont pas nécessairement locales au sens des distances des projections sur les différents axes. Ceci est appelé le fléau de la dimension. Au-dessus de 3 variables d'entrée, la procédure LOESS commence à être affectée par le fléau de la dimension. Ceci peut mener la méthode à négliger certains effets de variables importantes dans l'estimation de f.
Pour surmonter ce problème, de nombreuses procédures ont été développées, faisant appel à une ou plusieurs des méthodes suivantes : modèles additifs (cf. Annexe B), réduction de la dimension (cf. Section 3) et partitionnement récursif (cf. Annexe B).

Modèles d'interpolation Le dernier type de méta-modèles que nous allons voir regroupe les méthodes interpolantes, c'est-à-dire les modèles qui passent par les points de l'échantillon. On parle donc d'une erreur d'apprentissage nulle. Différentes méthodes satisfont cette caractéristique :

- les polynômes d'interpolation,
- l'interpolation spatiale par krigeage,
- les réseaux RBF (Radial Basis Function).

Nous ne nous intéresserons qu'aux polynômes d'interpolation dans cette section. Les deux autres méthodes sont décrites en Annexe B.

Polynômes d'interpolation Nous expliquerons ici le cas des polynômes de Lagrange. Il est également possible d'utiliser des polynômes d'Hermite ou leur généralisation, les polynômes d'Hermite-Birkhoff. On dispose de $N+1$ points dans l'échantillon d'apprentissage $(x^0, y_0), \ldots, (x^N, y_N)$ avec des x^i distincts 2 à 2. Ces points sont appelés des nœuds. Le but de la méthode est de construire un polynôme de degré minimal qui, aux abscisses x^i prend les valeurs y_i. On obtient alors le polynôme de degré au plus N

$$\hat{f}(x) = \sum_{j=0}^{N} y_j l_j(x), \tag{II.26}$$

où les fonctions l_j sont les polynômes de Lagrange définis en dimension 1 ($d = 1$) comme

$$l_j(x) = \prod_{i=0, i\neq j}^{N} \frac{x - x^i}{x^j - x^i}.$$

Ces polynômes vérifient les deux propriétés suivantes :

- l_j est de degré N, pour tout j,
- $l_j(x^i) = \delta_{ij}$, pour tout $1 \leq i, j \leq N$, avec $\delta_{jj} = 1$ et $\delta_{ij} = 0$, pour $i \neq j$.

La modélisation en dimension supérieure est possible grâce au fait que l'interpolation sur d variables est le produit des interpolations marginales sur chacune des d variables, à condition que le domaine d'entrée soit borné. Considérons le cas bidimensionnel où nous disposons de $N+1$ valeurs pour une variable x_1 et $M+1$ valeurs pour une variable x_2. Notre échantillon d'apprentissage est donc constitué de $(N+1)(M+1)$ points. On considère 2 familles de polynômes de Lagrange :

$$L_j^1 = \prod_{i=0, i\neq j}^{N} \frac{x - x^i}{x^j - x^i}$$

et

$$L_j^2 = \prod_{i=0, i\neq j}^{M} \frac{x - x^i}{x^j - x^i}$$

En supposant que f est continue sur le domaine d'entrée, on obtient le polynôme d'interpolation de Lagrange de degré au plus N en x_1 et au plus M en x_2 de la forme :

$$\hat{f}(x_1, x_2) = \sum_{i=0}^{N} \sum_{j=0}^{M} f(x_1^i, x_j^2) L_i^1(x_1) L_j^2(x^2) \tag{II.27}$$

La complexité du modèle obtenu dépend du nombre de points dans l'échantillon d'apprentissage. Dans (Broniatowski and Celant, 2014), il est proposé de trouver le plan optimal de l'interpolation.

La formulation analytique du modèle par polynômes de Lagrange est plutôt simple et permet donc d'obtenir assez rapidement l'estimation de nouveaux points. La méthode permet également de traiter des cas de non-linéarités pas trop fortes. Par contre, l'application en multidimensionnel devient vite difficile avec autant de familles de polynômes qu'il y a de variables d'entrée. Enfin, autant la méthode présente une erreur d'apprentissage nulle, autant l'erreur de test peut être importante.

Dans un cas déterministe, comme c'est le cas avec les codes de calculs, quelques recommandations sont proposées dans (Simpson et al., 2001). Concernant la méthode d'échantillonnage, ils préconisent les plans « space-filling », comme les LHS, utilisable pour tous les types de méta-modèles. Ensuite, en ce qui concerne le choix du modèle, les auteurs distinguent trois cas :

- on a beaucoup de facteurs (>50) : préférer les réseaux de neurones même s'ils sont coûteux à créer,
- on a des tendances non linéaires et peu de facteurs (<50) : préférer le krigeage même si cela ajoute de la complexité,
- on a très peu de facteurs (<10) avec un bon comportement : préférer les surfaces de réponse polynomiales à faible degré.

Ici, nous avons présenté uniquement les méta-modèles que nous avons ensuite utilisés sur les cas tests. D'autres méta-modèles sont décrits en Annexe B. Le choix du type de méta-modèle peut se faire suivant plusieurs critères. Le plus important est la connaissance que nous avons du système. Ensuite, il y a la connaissance acquise avec les observations et les représentations graphiques. Puis il y a les caractéristiques du système comme le nombre de facteurs à prendre en compte. Si avec tous ces critères il reste plusieurs choix de modèles, il suffit d'en tester plusieurs et de les comparer grâce à des indicateurs de précision, décrits dans la section suivante.

4.2.3 Méthodes de validation du modèle

Les méta-modèles doivent être validés avant d'être utilisés comme approximation d'un code de calculs.

Dans (Jin et al., 2001) sont proposés 5 aspects selon lesquels la performance des techniques de méta-modélisation peut être mesurée :

- Précision : la capacité à prédire la réponse du système sur le domaine d'intérêt, autant sur l'échantillon d'apprentissage que sur l'échantillon de test.
- Robustesse : la capacité à atteindre une bonne précision pour des problèmes et des tailles d'échantillons différents.
- Efficacité : l'effort calculatoire requis pour construire le méta-modèle et pour prédire la réponse pour un ensemble de nouveaux points (échantillon de test).
- Transparence : la capacité à illustrer explicitement les relations entre les variables d'entrée et la réponse.
- Simplicité conceptuelle : la facilité d'implémentation. Des méthodes simples demandent un minimum d'interventions de l'utilisateur et sont faciles à adapter à chaque problème.

Dans (Villa-Vialaneix et al., 2012), il est également question d'efficacité, de temps de calculs et d'interprétation du méta-modèle.

Lorsque des points supplémentaires sont disponibles (on suppose que l'on en a m), la précision du modèle peut être déterminée grâce à 3 indicateurs :

- Le coefficient de détermination $R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = 1 - \frac{MSE}{Variance}$, où $\hat{y}_i$ est la valeur prédite correspondant à la valeur observée y_i, $\bar{y}$ est la moyenne des valeurs observées. Tandis que le MSE (erreur quadratique moyenne) représente l'écart du méta-modèle au code, la variance représente l'irrégularité du modèle. Plus le R^2 est grand, plus le méta-modèle est précis. Ce coefficient peut également être calculé sur les points de l'échantillon d'apprentissage.
- L'erreur relative moyenne absolue $RAAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N \times STD}$, où STD est l'écart-type. Plus la valeur de $RAAE$ est petite, plus le méta-modèle est précis.
- L'erreur relative maximum absolue $RMAE = \frac{\max(|y_1 - \hat{y}_1|, \ldots, |y_N - \hat{y}_N|)}{STD}$. De grandes valeurs du $RMAE$ indique une erreur importante dans une région du domaine bien que la précision générale indiquée par R^2 et $RAAE$ puisse être bonne. Une valeur faible de $RMAE$ est préférable mais comme cet indicateur ne montre pas la performance globale, il n'est pas aussi important que les 2 autres.

Si un échantillon de test est trop coûteux à obtenir, il est possible de procéder par validation croisée. Ainsi, un ou plusieurs points de l'échantillon d'apprentissage jouent tour à tour le rôle d'échantillon de test pendant que le reste de l'échantillon d'apprentissage est utilisé pour construire le méta-modèle. Le coefficient de détermination pour la validation croisée est $Q^2 = 1 - \frac{\sum_{i=1}^{N}(\hat{y}^{(N-m)} - y_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}^{(N-m)})^2}$, où $\hat{y}_i^{(N-m)}$ sont les prédictions avec le méta-modèle construit avec l'échantillon comprenant $N - m$ points et $\bar{y}^{(N-m)}$ la moyenne sur cet échantillon.

La mesure de la robustesse consiste à déterminer l'impact d'un comportement bruité des entrées sur le système. L'efficacité d'une technique de méta-modélisation est mesurée par le temps utilisé pour la construction du modèle et les nouvelles prédictions. Ce temps dépend de la complexité du problème, de la dimension des entrées et de la taille de l'échantillon d'apprentissage.

5 Réduction de la dimension et méta-modélisation pour le cas test principal

Le cas test principal, consistant à réaliser le dimensionnement aérodynamique et mécanique d'un compresseur HP (CHP), dispose de 7 entrées (les sorties du dimensionnement aérodynamique). Parmi toutes les sorties du système, nous nous intéressons à 2 sorties en particulier (la masse du CHP et la variable clash qui mesure la distance entre le palier 3 et la veine). Une dimension de 7 en entrée ne paraît pas trop grande mais le fait de vouloir accélérer et opti-

miser les temps de calculs des études nous pousse à réduire cette dimension. Cette réduction sera également utile pour effectuer les deux autres objectifs de la thèse, à savoir l'optimisation robuste de la masse et la résolution de problèmes d'intégration (variable clash négative) par inversion de fonctions. Nous serons également amenés à utiliser les méta-modèles pour remplir ces deux objectifs.
Nous considérons les deux sorties comme deux cas indépendants. Sur chacun s'effectuera la réduction de la dimension et la méta-modélisation.

5.1 Étude de la masse du CHP dans le logiciel R

La masse estimée du compresseur HP est la masse des différents composants du compresseur : les aubes et les disques. Il s'agit d'un calcul simplifié puisque le compresseur HP comprend beaucoup d'autres pièces, nous n'avons considéré que les principales.
Nous cherchons les entrées les plus influentes sur la masse, parmi les sept variables à notre disposition, notées de A à G. Ceci contribue à la compréhension du problème et permet ensuite de constituer un bon méta-modèle pour la masse.
Les sept entrées sont bornées, bornes choisies par les ingénieurs avant-projets et représentant $\pm 10\%$ des valeurs nominales.

5.1.1 Réduction de la dimension par criblage

Afin d'utiliser la méthode de criblage, nous avons réalisé un plan factoriel fractionnaire à 2 niveaux : un plan de Taguchi à 16 expériences, $L_{16}(2^{15})$. Les 2 niveaux de chaque variable sont leurs bornes. Ce plan est de résolution IV, il nous permet donc d'étudier les effets des facteurs sans confusion avec les interactions. Par contre, les interactions sont confondues entre elles.

Le problème qui se pose ici est que toutes les expériences du plan ne sont pas exploitables, ce sont des échecs. Les valeurs des entrées amènent à une condition non satisfaite dans un des outils utilisés pour le dimensionnement. La non-satisfaction de cette condition conduit à un échec du calcul qui ne s'effectue pas, toutes les sorties sont à 0. La réduction des bornes par une étude des contraintes n'a pas permis d'obtenir un domaine où toutes les expériences du plan étaient exploitables.
Si toutes les expériences ne sont pas exploitables, il est impossible d'exploiter le plan discret et d'appliquer la méthode de criblage car on perd la propriété d'orthogonalité du plan. Il nous faut donc utiliser une autre méthode de sélection de variables, comme la sélection de modèle.

5.1.2 Sélection de variables par sélection de modèle

Pour effectuer une sélection de modèle, il est nécessaire de disposer d'un ensemble d'observations bien réparties sur le domaine d'entrée. Pour cela, nous réalisons un plan latin hypercube sur le domaine de dimension 7, contenant 150 points. Les 7 dimensions correspondent aux 7 entrées A, B, C, D, E, F et G. Nous retirons les échecs de ce plan, ce qui nous donne un échantillon à 144 points. Sur ce plan, nous gardons aléatoirement trois quarts des points pour constituer l'échantillon d'apprentissage à 108 points et le quart restant est l'échantillon de test à 36 points. Il est également possible de générer deux plans séparément.
Le modèle complet d'ordre 1 (II.19) est obtenu dans R par la commande *lm*. Cette commande permet d'obtenir, à partir d'un échantillon, des modèles polynomiaux de tous ordres, dans la limite du nombre d'expériences de l'échantillon d'apprentissage. Le modèle complet obtenu possède un coefficient de détermination (R^2) de 0.137 sur l'échantillon d'apprentissage tandis que l'erreur quadratique moyenne (MSE) sur l'échantillon de test est de 355.06, ce qui est très élevé. La relation entre la masse du CHP et les entrées du système ne peut pas être fidèlement représentée par un modèle linéaire d'ordre 1. Le passage à l'ordre 2 (II.20) améliore la représentativité mais ne fournit pas de résultats satisfaisants avec $R^2 = 0.567$ et $MSE = 312.15$. L'ordre 3 n'est pas possible car le modèle comporte 120 termes et nous ne disposons que de 108 expériences. Testé avec l'échantillon complet à 144 expériences, le modèle d'ordre 3 a un $R^2 = 0.951$. La valeur élevée n'est due qu'au fait d'ajouter beaucoup de termes car $R^2_{adj} = 0.711$. Il n'est donc pas réaliste de considérer le méta-modèle de la masse comme un modèle polynomial. Cependant, nous pouvons tout de même nous servir de la sélection de modèles à l'ordre 1 pour réaliser une sélection de variables. L'ordre 2 n'est pas nécessaire car, pour la sélection de variables, seules les variables nous intéressent, pas les interactions ou les termes carrés. Comme nous ne nous en servons pas pour la méta-modélisation, il est inutile de considérer un échantillon de test. Nous utiliserons donc les 144 points pour effectuer cette sélection de variables.

Initialement, nous disposons de deux modèles : le modèle complet d'ordre 1 contenant tous les termes linéaires et une constante et le modèle nul contenant uniquement la constante. Nous effectuons une sélection de variables pas à pas grâce à la commande *step* de R. Pour cela, nous comparons les résultats des recherches progressive (ajout de termes à partir du modèle nul), régressive (retrait de termes à partir du modèle complet) et séquentielle (ajout ou retrait à chaque itération). Dans les trois cas, on cherche à minimiser un critère de type AIC ou BIC. Pour chacune des trois méthodes et avec le critère AIC, les variables A, C et G sont sélectionnées. Avec le critère BIC, seule la variable A est sélectionnée, quelle que soit la méthode.

Nous comparons également ces trois méthodes à une quatrième, la recherche exhaustive qui consiste à comparer tous les modèles possibles. Elle s'obtient dans R par la commande *regsubsets* du package *leaps*. La comparaison est basée sur la minimisation du Cp de Mallows et est représentée à la Figure II.10. Chaque graphique correspond à une des quatre méthodes. Sur chacun, on dispose de 7 modèles, contenant de une à sept variables. Sur un graphique, chaque ligne correspond à un modèle. Un carré noir signifie que la variable en abscisse est présente dans ce modèle selon le critère choisi. En ordonnées, on a la valeur du critère.

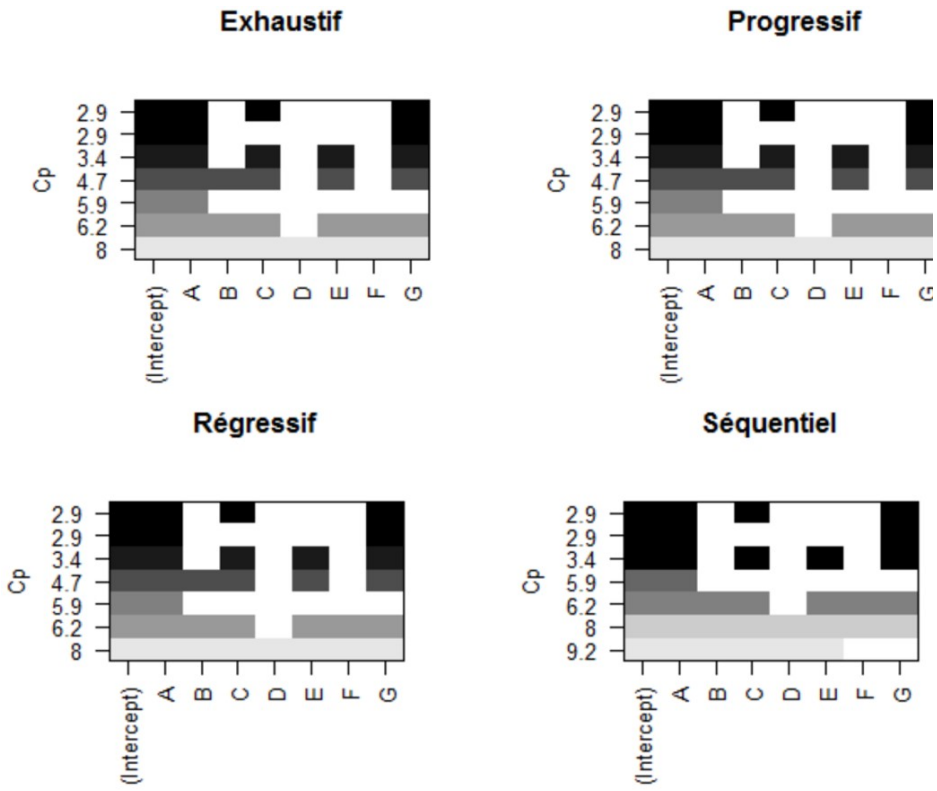

Figure II.10 – Comparaison de quatre méthodes de sélections de variables pour la masse suivant le critère du Cp de Mallows

La Figure II.10 montre que les quatre méthodes donnent pratiquement les mêmes résultats. Celui qui minimise le Cp de Mallows (ligne les plus haute sur chaque graphique) est le modèle contenant les variables A, C et G.

Pour la méthode exhaustive, nous comparons les résultats pour différents critères : Cp de Mallows, R^2, R^2 ajusté et BIC. Les critères Cp et BIC sont à minimiser, les deux autres sont à maximiser. Ces résultats sont présentés sur la Figure II.11 dont les graphiques se lisent de la même façon que la figure précédente.

On voit sur cette figure que les résultats diffèrent selon le critère choisi. Cependant, le critère

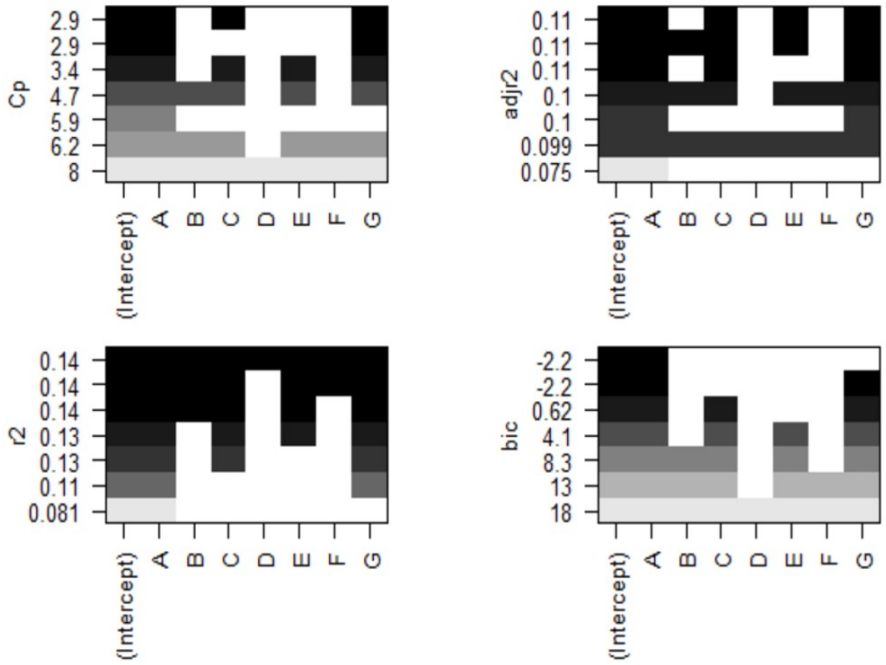

Figure II.11 – Comparaison des résultats de la méthode exhaustive de sélection de variables pour la masse par rapport à quatre critères

Cp conduit à la sélection des variables A, C et G comme les méthodes précédentes et comme le critère AIC. Le critère BIC ne sélectionne que la variable A. Le critère R^2 sélectionne les mêmes variables que précédemment mais en ajoute quatre. En fait, il prend en compte toutes les variables. Ceci est dû au fait que le R^2 a tendance à prendre le maximum de termes puisque sa valeur augmente avec le nombre de termes. Le critère R^2 ajusté propose trois modèles avec la même valeur du critère. Parmi ces modèles, on retrouve celui contenant A, C et G.

Nous choisissons donc de garder les entrées A, C et G pour représenter la masse du CHP. L'étape suivante, après la sélection de modèle, consiste à déterminer un méta-modèle pour la masse.

5.1.3 Justification de l'absence de méta-modèle pour la masse

Dans un but d'optimisation de la masse du CHP, dans des temps de calculs raisonnables, nous avons d'abord réduit la dimension du problème afin de limiter la taille du domaine à explorer et donc le nombre d'expériences nécessaires à toute étude. De sept entrées, nous sommes ainsi passés à trois. L'idéal pour réduire les temps de calculs est de travailler sur un méta-modèle plutôt que sur le code de calculs coûteux. Le problème ici est que nous disposons également

d'un certain nombre de contraintes en sortie du système. L'optimum en masse doit satisfaire toutes ces contraintes. La méthode d'optimisation sous contraintes ne peut donc être effectuée que sur le code de calculs, à moins d'avoir un méta-modèle pour chaque contrainte. Ceci serait d'un côté fastidieux car nous disposons de 10 contraintes et d'un autre côté risqué car l'erreur de modélisation peut conduire à un optimum satisfaisant les contraintes selon les méta-modèles mais qui ne les satisfait pas en réalité.
L'optimisation de la masse, sous la condition de satisfaire les 10 contraintes du système, est donc effectuée directement sur le code de calculs sur le domaine de dimension 3, correspondant aux domaines de variations de A, C et G.

5.2 Étude du clash entre le palier 3 et la veine

Le clash entre le palier 3 et la veine est la distance, en millimètres, entre le coin en haut à droite du palier 3 et la partie de la veine la plus proche du palier. Le clash est représenté à la Figure II.12.

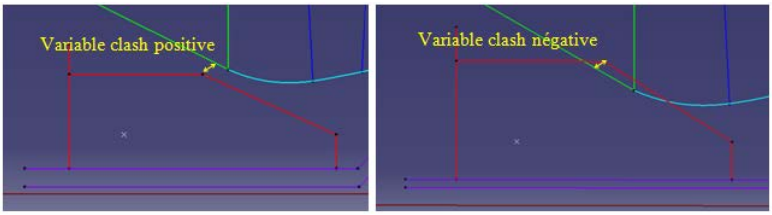

Figure II.12 – Représentation de la sortie clash sur le dimensionnement du CHP

A gauche, nous sommes dans une configuration où le clash est positif, il n'y a donc pas de collision entre le palier 3 et le compresseur. A droite, il y a un problème d'intégration car le dimensionnement conduit à un clash négatif traduisant une collision.

Comme pour l'étude de la masse, les échecs parmi les expériences nous empêchent d'effectuer une sélection de variables par criblage. Nous allons donc utiliser la sélection de modèle et l'analyse de sensibilité.
L'objectif est le même que précédemment, à savoir trouver les entrées les plus influentes parmi les sept variables dont nous disposons. La différence est que la sortie d'intérêt n'est plus la masse mais le clash.

	R^2	R^2 ajusté	MSE	$RAAE$	$RMAE$
Modèle complet d'ordre 1	0.9695	0.9673	53.35	0.139	0.531
Modèle d'ordre 1 avec interactions	0.9907	0.9874	24.45	0.096	0.304
Modèle complet d'ordre 2	0.995	0.9925	27.86	0.073	0.541

Tableau II.5 – Comparaison des qualités de trois modèles complets pour le clash

5.2.1 Sélection de variables par analyse de sensibilité

L'analyse de sensibilité peut être effectuée via un méta-modèle sur la sortie d'intérêt. Ce modèle permet d'estimer les indices de Sobol d'ordre 1 qui nous donneront l'influence de chaque entrée sur la sortie. Nous reprenons l'échantillon à 144 points utilisé pour la masse. Nous le séparons en un échantillon d'apprentissage représentant 3/4 des points, le reste constituant l'échantillon de test.

A partir de l'échantillon d'apprentissage, on établit un modèle. Pour cela, nous comparons le modèle complet d'ordre 1, le modèle d'ordre 1 avec interactions et le modèle complet d'ordre 2 selon plusieurs critères. Ces comparaisons sont répertoriées dans le Tableau II.5. Les critères R^2 et R^2 ajusté à maximiser sont calculés sur l'échantillon d'apprentissage. Les critères MSE, $RAAE$ et $RMAE$ à minimiser sont calculés sur l'échantillon de test. On voit bien que ces trois modèles ont de bonnes qualités de représentation et de prédictivité. Le clash pourra donc être représenté par un modèle polynomial. Sachant cela, nous pourrions donc effectuer la sélection de variables en même temps que la méta-modélisation par les méthodes de sélection de modèle. C'est ce que nous ferons dans les sections suivantes. Nous pourrions utiliser le méta-modèle ainsi obtenu pour calculer les indices de Sobol. Mais une fois la sélection de modèle effectuée, le calcul des indices de Sobol n'est qu'une confirmation des résultats obtenus. C'est pourquoi nous allons les calculer sur un modèle complet, qui n'est pas le modèle optimal mais qui est de bonne qualité. Ainsi, nous pourrons comparer les résultats obtenus par analyse de sensibilité et ceux issus de la sélection de modèle.

Les indices de Sobol de la Figure II.13 ont été calculés dans Optimus à partir du modèle complet d'ordre 2.

Ces indices montrent que les entrées les plus influentes sur le clash sont successivement : A, G, F et B. Les trois autres ont des effets nuls sur la sortie. Les chiffres entre crochets sur la Figure II.13 sont les effets totaux.

Avec les autres modèles, les indices n'ont pas exactement les mêmes valeurs mais on en tire les mêmes conclusions sur le classement des entrées selon leur effet sur le clash. Dans R, de nombreuses fonctions du package *sensitivity* permettent de calculer les indices de Sobol d'ordre

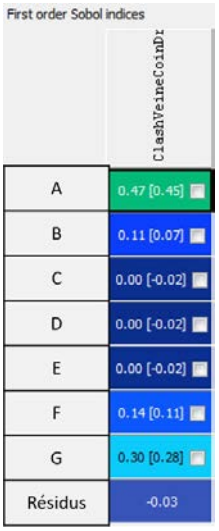

Figure II.13 – Valeurs des indices de Sobol d'ordre 1 pour le clash

1 et les indices totaux. Sur le modèle complet d'ordre 1, on obtient des indices que l'on représente à la Figure II.14.

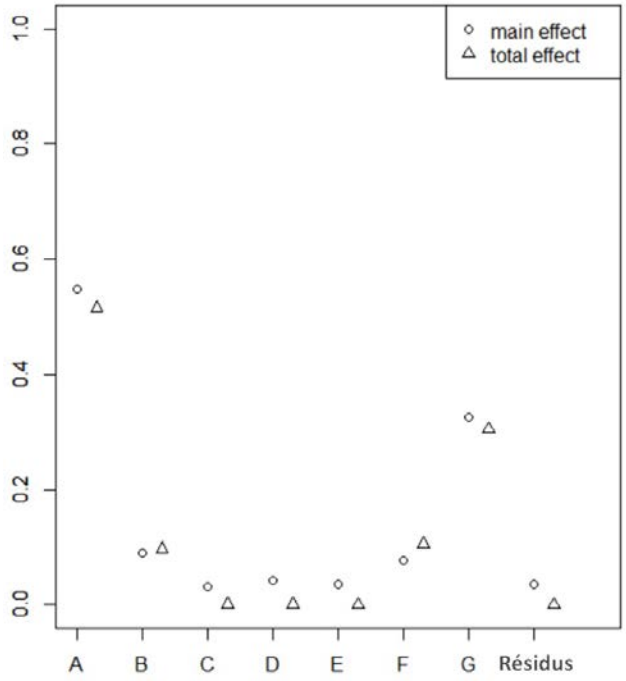

Figure II.14 – Représentation des indices de Sobol d'ordre 1 et totaux pour le clash

Les indices d'ordre 1 sont représentés par des cercles, les indices totaux par des triangles. On conclut comme précédemment que les entrées les plus influentes sont A, G, F et B. Le seul

changement est que l'ordre des deux dernières entrées est inversé.
Nous confirmons ces résultats par la sélection de modèle qui va permettre également d'établir un méta-modèle pour le clash.

5.2.2 Sélection de variables par sélection de modèles et établissement du méta-modèle

Pour appliquer les méthodes de sélection de modèle, nous établissons un modèle vide contenant uniquement une constante et un modèle complet. Suivant les critères du Tableau II.5, nous choisissons le modèle complet d'ordre 2.
Notons que l'étude sur le modèle complet d'ordre 1, menée comme pour la masse sur les critères AIC et BIC, a conduit à la sélection des quatre mêmes entrées que les indices de Sobol.

Le modèle complet d'ordre 2 comporte 36 termes. Nous voulons effectuer une sélection sur ces 36 termes. Parmi eux, tous les termes relatifs aux entrées C, D et E sont écartés, les autres sont tous conservés, comme le montre les encadrés rouges de la Figure II.15. En abscisse, on trouve les termes du modèle à 7 variables. Le terme principal de la variable A s'écrit 1.0.0.0.0.0.0 car c'est la première variable, celui de la variable C s'écrit 0.0.1.0.0.0.0 car c'est la troisième. L'interaction entre A et C s'écrit 1.0.1.0.0.0.0. Le terme carré de A s'écrit 2.0.0.0.0.0.0. Les résultats de la sélection de modèle par recherches exhaustive et progressive selon le critère Cp sont représentés. Il s'agit donc du modèle complet d'ordre 2 à 4 variables. Ce modèle possède

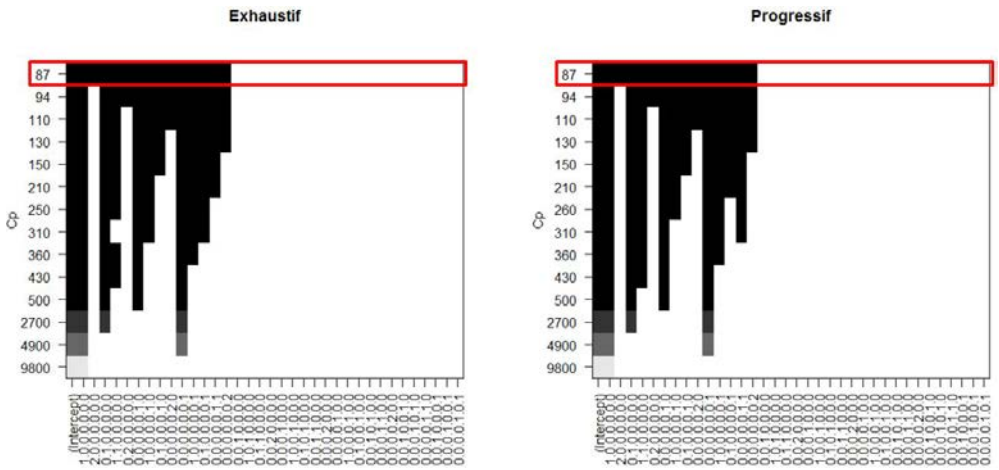

Figure II.15 – Sélection de modèle pour le clash

de bonnes qualités de représentativité : $R^2 = 0.991$ et $R^2_{ajusté} = 0.99$. La représentativité est meilleure que pour le modèle d'ordre 1 avec interactions mais légèrement moins bonne que le

modèle complet d'ordre 2 qui possède plus de termes. Il a également de bonnes qualités prédictives : $MSE = 20.84$, $RAAE = 0.069$ et $RMAE = 0.49$. Les critères MSE et $RMAE$, qui sont des critères de prédictivité globale du modèle, sont meilleurs par rapport à ceux des autres modèles du Tableau II.5. Le critère $RMAE$, qui est un critère local, est moins bon que celui du modèle d'ordre 1 avec interactions. Notre modèle a donc des erreurs maximales plus grandes mais en moyenne, la prédictivité est meilleure. Nous conserverons donc ce modèle dans les prochaines études menées sur le clash.

La régularité de la sortie « clash » nous permet d'utiliser des méthodes usuelles de méta-modélisation. Nous allons voir avec le cas test secondaire de l'effort au palier que ce n'est pas toujours le cas. En effet, il est parfois nécessaire d'utiliser des méthodes plus personnalisées suivant le problème étudié.

6 Méta-modélisation pour le cas test secondaire

Le cas test secondaire porte sur la calibration de jauges et l'évaluation de l'effort à partir de signaux de jauges. La phase qui nous intéresse ici est la calibration puisque c'est l'étape expérimentale où il faut établir le modèle qui sera ensuite utilisé pour l'évaluation de l'effort. Comme cela a été décrit au Chapitre I, la phase de calibration consiste à mesurer les déformations subies par un palier et causées par un certain effort appliqué sur ce palier. Cet effort est caractérisé par deux variables : la force qui varie de 0 à 26000 daN et l'angle qui prend 6 valeurs, comme on peut le voir à la Figure II.16. Les déformations sont mesurées par quatre jauges ($S15$, $S30$, $S45$ et $S60$) équi-réparties sur le palier (cf. Figure II.16).

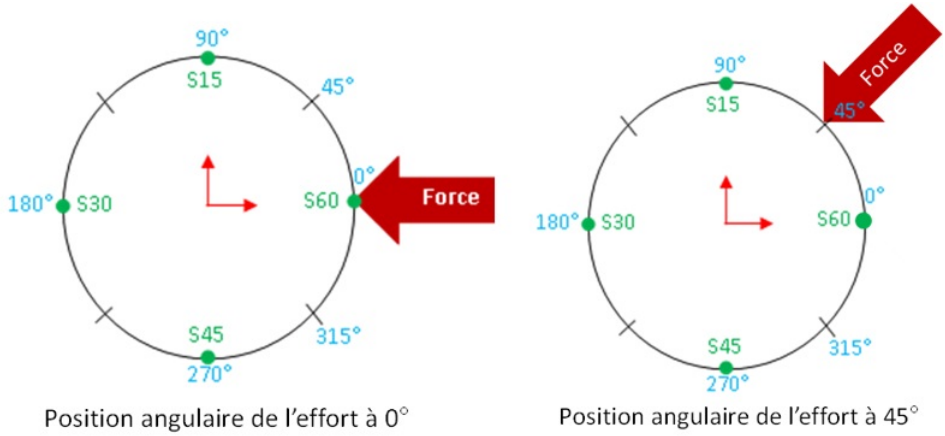

Figure II.16 – Représentation des essais statiques sur le palier 1 à 4 jauges

Le méta-modèle, appelé ici fonction de transfert, doit permettre d'exprimer la relation entre la variable représentant les déformations, considérée comme la réponse, et le couple (force,angle), qui sont alors vues comme les entrées.
Nous disposons de deux types de données :

- des données de calibration fournissant les déformations mesurées en chaque jauge en fonction des positions angulaires de l'effort, pour une force fixée,
- des données de traction fournissant pour chaque angle les déformations mesurées en chaque jauge en fonction de la force.

En pratique, la déformation est supposée linéaire par rapport à la force. Si cette relation présente de fortes non-linéarités, alors les jauges sont jugées inacceptables et doivent être vérifiées. Pour les données de traction dont nous disposons, la relation est effectivement linéaire à partir de 5000 daN, avant ce n'est pas forcément le cas. Nous pouvons représenter cette relation entre la déformation et la force, par angle (6 graphiques) et par jauge (4 courbes par graphique). La Figure II.17 représente les courbes obtenues pour des angles de 0 ° et 45 ° . On remarque que l'on obtient les mêmes courbes à 0 ° , 90 ° , 180 ° et 270 ° , seules les couleurs sont interverties. Ceci est dû à l'axisymétrie du palier qui assure que les jauges sont interchangeables. On trouve également les mêmes courbes pour 45 ° et 315 ° .

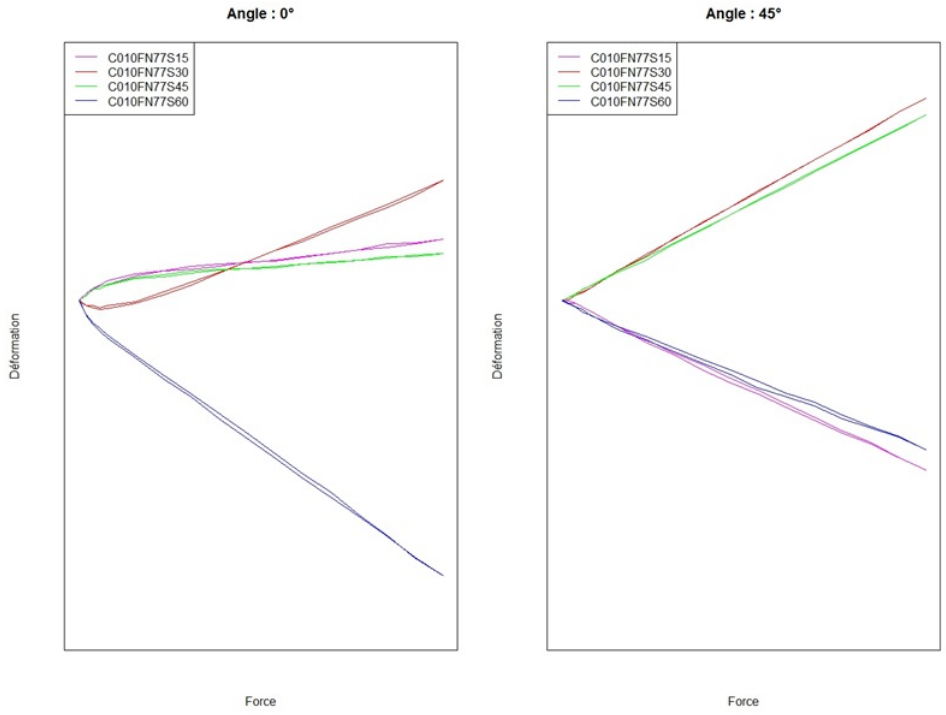

Figure II.17 – Représentation des données de traction par angle de l'effort

Position angulaire de l'effort	Déphasage $S60$	Déphasage $S15$	Déphasage $S30$	Déphasage $S45$
0	0	90	180	270
45	315	45	135	225
90	270	0	90	180
180	180	270	0	90
270	90	180	270	0
315	45	135	225	315

Tableau II.6 – Déphasage par position angulaire de l'effort et par jauge

Dans un premier temps, nous considérerons donc que les déformations évoluent linéairement en fonction de la force. Cette force est fixée à 20000 daN (= 20 tonf), valeur pour laquelle nous possédons les données de calibration. Ceci nous permet d'établir la fonction de forme pour une force fixée. Ensuite, nous ferons varier la force.
Pour un effort appliqué selon une certaine position angulaire, chaque jauge forme un angle appelé déphasage (cf. Figure II.16) avec cette position. Par exemple, lorsque l'effort est appliqué sur le palier à 0 ° , le déphasage entre les jauges et l'effort est de 0 ° pour $S60$, 90 ° pour $S15$, 180 ° pour $S30$ et 270 ° pour $S45$. Ce déphasage est une quantité positive, il s'agit de la différence entre la position angulaire de la jauge et celle de l'effort, modulo 360. Les déphasages des quatre jauges pour chacune des positions angulaires sont fournis au Tableau II.6. La considération du déphasage de l'effort plutôt que de sa position angulaire permet de superposer les observations en chaque jauge, comme on peut le voir à la Figure II.18.

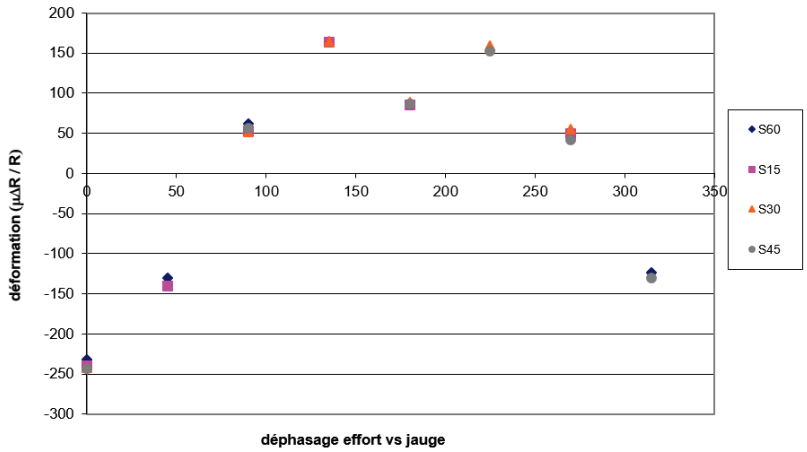

Figure II.18 – Représentation des données de calibration

La fonction proposée initialement était une fonction trigonométrique d'ordre 2

$$\varepsilon = F_0[c_0 + c_1 \cos(\varphi - \phi_1) + c_2 \cos(2\varphi - \phi_2)], \tag{II.28}$$

où ε est la déformation, F_0 la force de l'effort (ici 20 tonf), φ le déphasage et c_0, c_1, c_2, ϕ_1 et ϕ_2 les coefficients du modèle. Il s'agit donc d'un modèle paramétrique non-linéaire. Les cinq coefficients du modèle sont obtenus par moindres carrés. En chaque jauge, l'erreur commise par le modèle par rapport aux observations est

$$(Y_j - J_j)^2, \tag{II.29}$$

où Y_j est la déformation estimée pour la jauge j et J_j le signal mesuré en cette même jauge. Le choix des coefficients doit assurer que l'erreur totale soit minimale

$$E_{tot} = \sum_{j=1}^{4} (Y_j - J_j)^2. \tag{II.30}$$

A partir des données de calibration, on obtient une fonction représentée à la Figure II.19. Cette fonction n'est pas très bien ajustée, surtout pour les plus grandes déformations, l'erreur

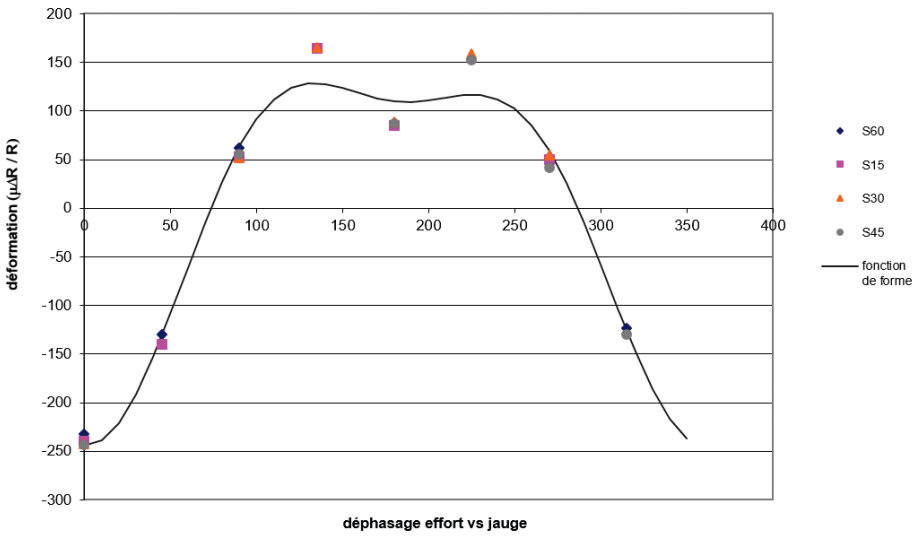

Figure II.19 – Fonction de transfert trigonométrique d'ordre 2

quadratique moyenne est d'ailleurs très grande.
Une première proposition a été faite pour obtenir un modèle de meilleure qualité. Il s'agit d'une somme de deux exponentielles. Cette idée est venue du fait que l'allure de la fonction fait penser

à la densité d'un mélange de deux lois normales. Le modèle est alors le suivant

$$\varepsilon = F_0 \left[\beta_0 + \beta_1 \left(e^{-\frac{1}{2}\left(\frac{\varphi-\beta_2}{\beta_3}\right)^2} + e^{-\frac{1}{2}\left(\frac{\varphi-\beta_4}{\beta_5}\right)^2} \right) \right]. \tag{II.31}$$

Les six coefficients de ce modèle sont également déterminés par moindres carrés, ce qui donne le graphique de la Figure II.20. On obtient un meilleur ajustement général avec une erreur

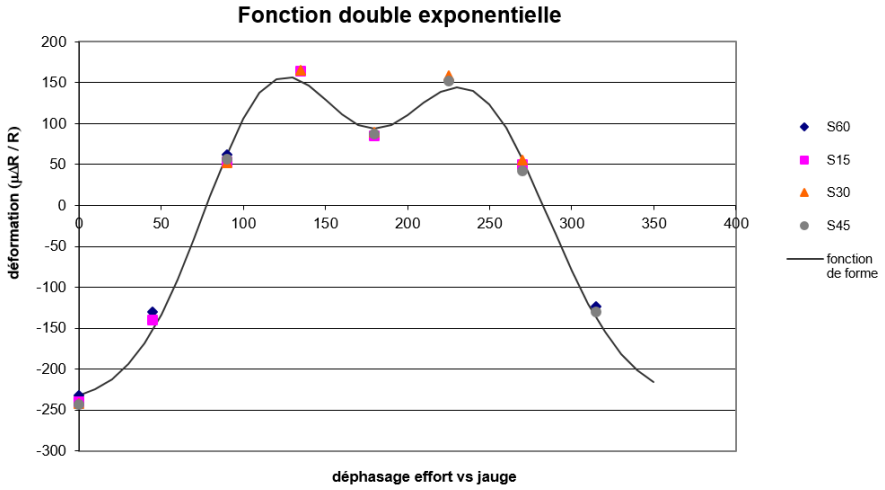

Figure II.20 – Fonction de transfert exponentielle

quadratique moyenne divisée par 4 par rapport au modèle précédent.
Le problème de ce modèle est que la fonction n'est pas périodique. En effet, la dérivée en 0 est différente de celle en 360, condition essentielle pour que la fonction présente une période de 360. De plus, la fonction doit être symétrique par rapport à 180.
La forme trigonométrique est donc plus adaptée. Pour améliorer la qualité du modèle, nous augmentons l'ordre. L'ordre 3 n'étant pas suffisant, nous sommes passés à 4

$$\varepsilon = F_0[c_0 + c_1\cos(\varphi - \phi_1) + c_2\cos(2\varphi - \phi_2) + c_3\cos(3\varphi - \phi_3) + c_4\cos(4\varphi - \phi_4)]. \tag{II.32}$$

On passe alors de 5 à 9 coefficients. Il apparaît évident que l'augmentation de l'ordre a des limites, il ne peut pas dépasser le nombre d'observations.
On évalue les coefficients par moindres carrés pour obtenir la fonction de la Figure II.21. On obtient un bon ajustement des observations avec une erreur quadratique moyenne divisée par 6 par rapport au modèle d'ordre 2. Parmi les 3 modèles proposés, nous choisissons donc celui-ci.
Le problème du calcul d'un modèle à partir de données d'essais est que nous avons souvent

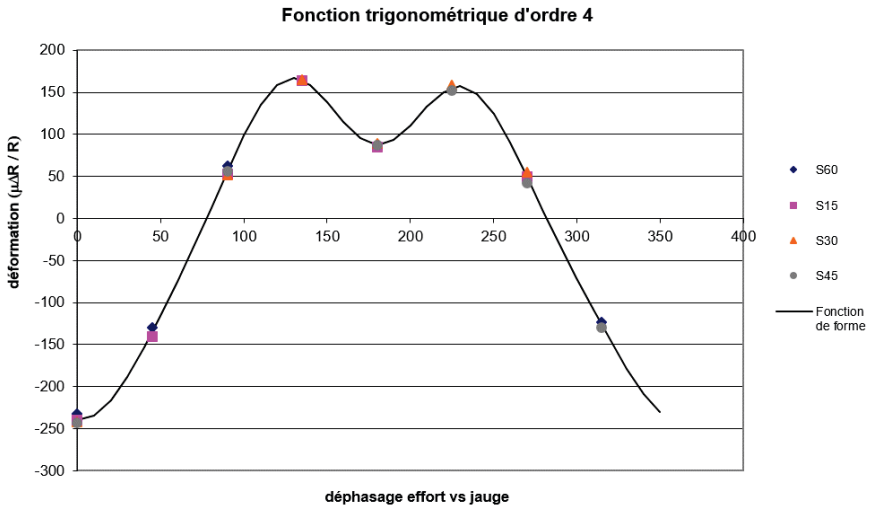

Figure II.21 – Fonction de transfert trigonométrique d'ordre 4

peu d'observations et que nous ne pouvons pas bénéficier d'un échantillon de test. En effet, il aurait été intéressant d'avoir des observations pour d'autres angles afin de tester les qualités prédictives du modèle. Une détermination de ces qualités par validation croisée n'est pas non plus possible car, pour certains déphasages, une seule observation est disponible. Retirer cette observation dégraderait donc grandement le modèle, ce qui fausserait le calcul d'indices de prédictivité.
Il est également inenvisageable de considérer une fonction de transfert par jauge, le nombre d'observations pour chacune des jauges n'est pas suffisant. Il faudrait ajouter au moins deux positions angulaires supplémentaires.

Pour ce type de fonctions de transfert, le modèle évolue de manière linéaire par rapport à la force puisqu'elle est mise en facteur dans le modèle. Nous représentons à la Figure II.22 la fonction trigonométrique d'ordre 4 avec la prise en compte d'une seconde dimension, la force. La quantité F_0 dans l'Equation II.33 est tout simplement remplacée par la variable force. Les deux graphiques de la Figure II.22 sont deux prises de vue de la fonction de transfert bidimensionnelle (en vert) qui ajuste les observations (représentées par des points). On voit bien que l'ajustement est bon, sauf pour les plus petites forces où la linéarité n'est pas forcément assurée.

Pour améliorer le modèle sur les petites forces, nous proposons de prendre en compte la force directement dans la construction de la fonction de transfert. Ceci permet d'obtenir un modèle qui n'est plus forcément linéaire en la force. L'idée première pour ajuster un modèle aux obser-

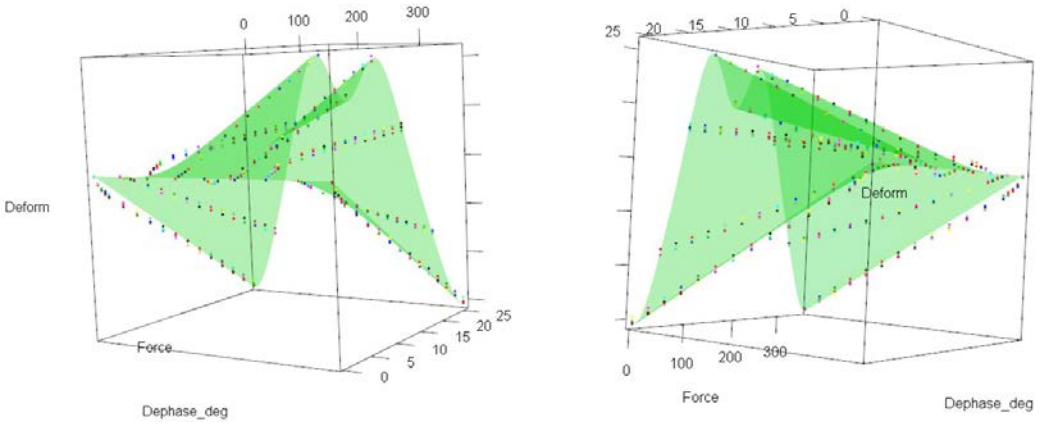

Figure II.22 – Fonction de transfert trigonométrique d'ordre 4 par rapport au déphasage et à la force

vations tracées à la Figure II.22 fut de considérer un modèle d'interpolation de type polynômes de Lagrange en dimension 2. L'ajustement était parfait mais le critère de périodicité n'était pas respecté. Nous sommes ici obligés de conserver la forme trigonométrique pour la variable position angulaire. Nous avons donc choisi de garder la forme trigonométrique et de faire évoluer les coefficients du modèle en fonction de la force. Le modèle trigonométrique d'ordre 4 devient

$$\begin{aligned}\varepsilon = F[&c_0(F) + c_1(F)\cos(\varphi - \phi_1(F)) + c_2(F)\cos(2\varphi - \phi_2(F)) + c_3(F)\cos(3\varphi - \phi_3(F)) \\ &+ c_4(F)\cos(4\varphi - \phi_4(F))],\end{aligned} \tag{II.33}$$

où F désigne la force. Les 9 coefficients du modèle sont évalués par moindres carrés pour chaque force grâce aux données de traction. Pour deux forces successives, il est clair que les coefficients du modèle ne sont pas indépendants. Nous pouvons donc rechercher les coefficients pour la plus grande force à notre disposition, 26 tonf, puis nous servir de ce résultat comme point de départ de la recherche des coefficients pour la force suivante. Notons que tous les coefficients évoluent dans $\mathbb{R}$ à l'exception des quatre coefficients situés dans les cosinus, qui évoluent dans $[-\pi, \pi]$. Chaque coefficient est représenté par rapport à la force (points noirs sur la Figure II.23) et on y ajuste un modèle non-linéaire en puissances de logarithme (courbes rouges sur la Figure II.23). On remarque qu'à partir de 5 tonf les coefficients ont une évolution linéaire, ce qui coïncide avec la linéarité observée sur les données de traction à partir de 5 tonf (cf. Figure II.17). Dans l'équation II.33, chaque coefficient est remplacé par le modèle logarithmique en F.

Cette méthode nous permet d'obtenir un modèle beaucoup plus fidèle aux observations des plus petites forces. Le problème est que ce modèle a également vocation à être utilisé pour l'extrapolation. En effet, les forces observées lors des essais statiques ne dépassent pas 26 tonf,

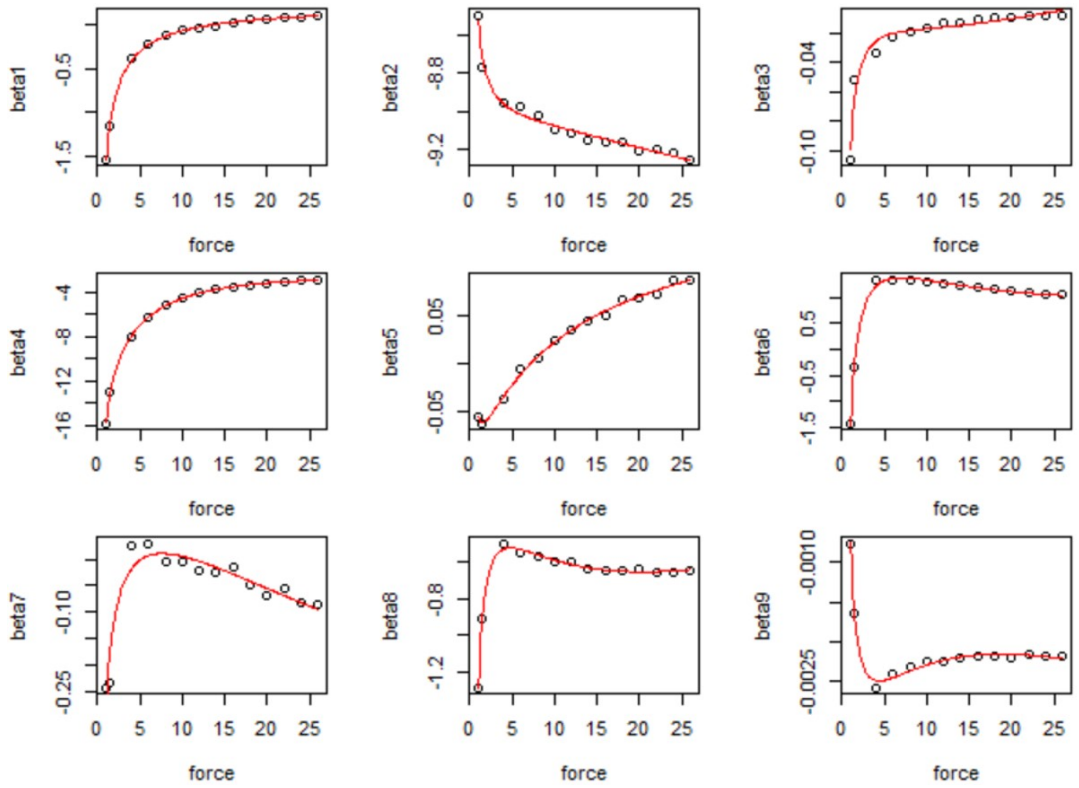

Figure II.23 – Représentation des coefficients du modèle trigonométrique d'ordre 4 en fonction de la force

tandis que lors d'essais réels en rotation, ces forces sont beaucoup plus importantes, pouvant atteindre 100 tonf sur notre cas d'application. Cet ajustement sur les petites forces risquent de détériorer le modèle en extrapolation.

Comme on a pu vérifier que le modèle était linéaire à partir de 5 tonf, nous réalisons une extrapolation linéaire suivant la force en ajoutant plusieurs points entre 26 tonf et 100 tonf. Nous ajoutons 7 points de 40 à 100 tonf et nous obtenons l'évolution des coefficients du modèle par rapport à la force avec ces points supplémentaires à la Figure II.24. Nous obtenons le modèle représenté à la Figure II.25. Il semble bien ajuster les points d'observation. Le modèle obtenu divise l'erreur quadratique moyenne par 8.6. La qualité du modèle peut être vérifiée par le tracé du graphe réel contre observé de la Figure II.26. Plus les valeurs estimées de la déformation grâce au modèle sont proches des valeurs réelles, plus les points du graphe sont alignés sur la bissectrice. L'erreur faible entre les données estimées et les données réelles est confirmée par ce graphe. La fonction de transfert ainsi établie représente un bon ajustement des données et pourra être utilisée ensuite pour évaluer l'effort à partir des signaux de jauges mesurés lors des essais en rotation.

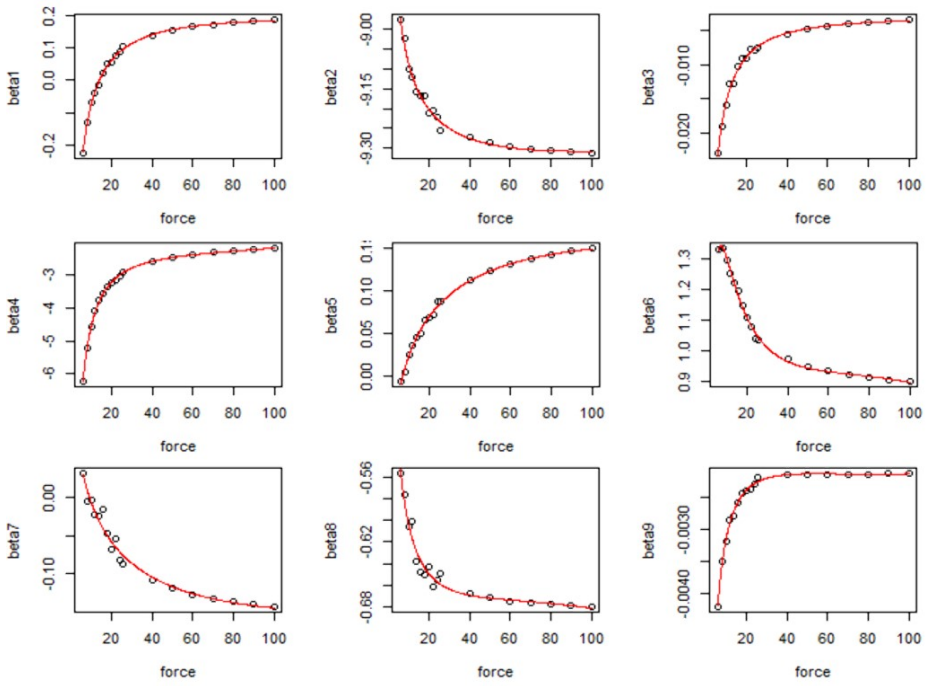

Figure II.24 – Représentation des coefficients du modèle trigonométrique d'ordre 4 en fonction de la force avec ajout de points d'extrapolation

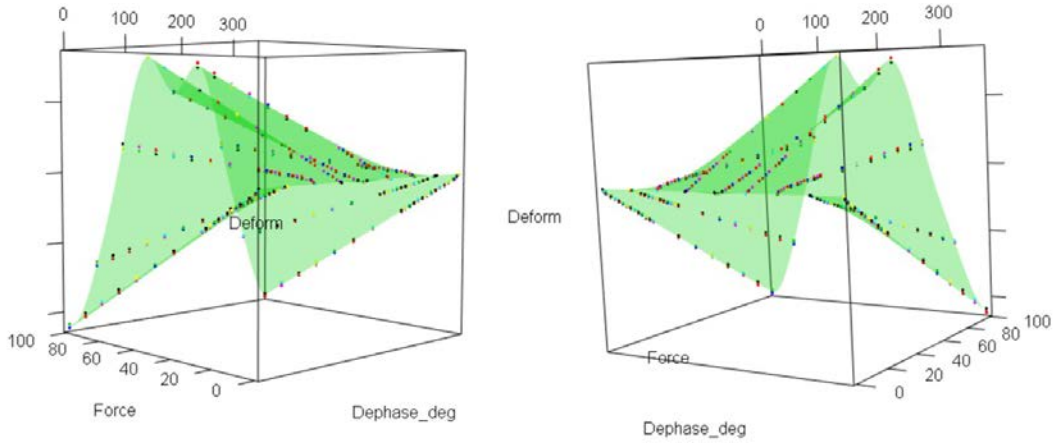

Figure II.25 – Fonction de transfert trigonométrique d'ordre 4 avec prise en compte de la force et de points d'extrapolation par rapport au déphasage et à la force

Le modèle que nous retenons pour cette étape de calibration de jauges est un modèle trigonométrique d'ordre 4 avec prise en compte de la force dans l'établissement des coefficients et ajout de points d'extrapolation. La construction de ce modèle n'est pas des plus standards. En effet, la forme de la fonction était imposée suivant une des deux variables, la position angulaire

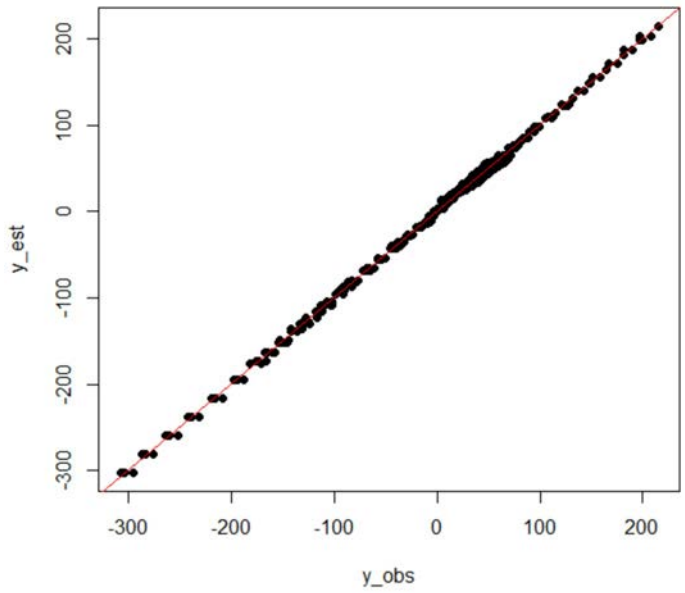

Figure II.26 – Graphique réel contre observé pour le modèle trigonométrique d'ordre 4 avec prise en compte de la force et ajout de points d'extrapolation

de l'effort. Ce modèle est donc propre à cette situation, difficilement transposable à des cas autres que l'effort aux paliers.

7 Conclusion

Nous avons vu dans ce chapitre qu'il existait de nombreuses méthodes de réduction de la dimension et de méta-modélisation. Certaines sont d'ailleurs complémentaires, d'autres permettent de remplir ces deux objectifs simultanément. Chaque méthode s'applique à certains types de problèmes et dans certaines situations. Il est donc indispensable de bien décrire le problème et l'objectif de l'étude. Nous avons notamment pu constater sur le cas test principal que le criblage ne pouvait pas s'appliquer lorsque certaines expériences sont des échecs. Il faut alors appliquer d'autres méthodes de réduction de la dimension comme la sélection de modèle ou les indices de Sobol.

Les méthodes de méta-modélisation dépendent également du type de problème. Les méthodes usuelles peuvent être appliquées dans les cas les plus réguliers. Dans des cas plus particuliers comme pour le cas test secondaire, il faut faire appel à des méthodes moins traditionnelles assurant les conditions du problème.

Une fois la réduction de la dimension effectuée et le méta-modèle obtenu, nous pouvons effectuer les principales études du problème. Il s'agit notamment de l'optimisation et de la résolution

de problèmes inverses. Ces deux objectifs font l'objet des deux chapitres suivants. Le premier porte sur l'optimisation robuste dont les méthodes seront appliquées au cas test principal pour optimiser la masse du CHP.

Chapitre III

Définition d'une méthodologie de conception robuste

Dans ce chapitre, nous nous intéressons à la formulation et aux méthodes de résolution d'un problème d'optimisation robuste. Ce problème diffère d'un problème d'optimisation simple (ou déterministe) dans le fait que l'aléa contenu dans le système est pris en compte dans l'optimisation. Ce sont des incertitudes, présentes à différents endroits du système et qui peuvent affecter plus ou moins les sorties de ce système. Ainsi, le résultat d'une optimisation peut s'avérer être très sensible à des petites variations sur les entrées par exemple. On dit alors que l'optimum n'est pas robuste. Les méthodes d'optimisation robuste consistent à déterminer la robustesse à chaque itération de l'algorithme. Ajoutée comme une contrainte, voire comme un objectif de l'optimisation, la robustesse fait totalement partie du processus et permet d'aboutir directement à un optimum robuste.

Les deux premières sections de ce chapitre sont consacrées aux incertitudes. Nous décrirons les étapes successives du traitement des incertitudes qui sont : le recensement, la modélisation et la propagation. La section suivante portera sur la formalisation du problème de conception robuste, qui est celle en vigueur à Safran Aircraft Engine. Nous exposerons ensuite plusieurs méthodes de résolution pour un tel problème de conception robuste. Enfin, la dernière section sera consacrée à l'application des méthodes d'optimisation robuste sur le cas test principal. L'objectif est d'obtenir un optimum robuste de la masse du compresseur HP, sous la satisfaction des contraintes du système.

1 Recensement des incertitudes

Rappelons d'abord quelques notations de la section 2 du Chapitre I. Nous disposons d'une boîte blanche dont les entrées $X = (X_1, ..., X_n)$ regroupent les entrées du système U et les paramètres environnementaux V. Les sorties sont représentées par $Y = (Y_1, ..., Y_k)$ et les contraintes par $T = (T_1, ..., T_m)$. Le code de calcul f tel que $Y = f(X)$ est la fonction objectif. Les fonctions g_i telles que $T_i = g_i(X), i = 1, ..., m$ sont les fonctions contraintes. Ces fonctions contraintes permettent de définir l'espace de conception par les relations $T_i \geq 0, i = 1, \ldots, m$.

1.1 Différents types d'incertitudes

Tout système physique, et donc notre boîte blanche, est sujet à différentes sources d'incertitudes qui sont en général classées en deux catégories (Beyer and Sendhoff, 2007) :

- *incertitudes objectives ou aléatoires* : incertitudes irréductibles dues à des phénomènes physiques non contrôlables comme la température, la pression mais aussi les conditions de fabrication de pièces ou les propriétés matériaux ;
- *incertitudes de modèle ou épistémiques* : incertitudes dues à un manque de connaissance du concepteur vis-à-vis du comportement de son système. Ces incertitudes peuvent être en général réduites par un effort de modélisation, d'investissement dans la connaissance.

Le vecteur X défini précédemment est entaché d'incertitudes aléatoires. Les incertitudes épistémiques touchent quant à elles les fonctions f et $g_i, i = 1, ..., m$.
Un classement plus précis des différentes variables incertaines en fonction de la possibilité ou non d'action sur ces variables dans le processus de conception a été proposé dans (Beyer and Sendhoff, 2007) :

- *incertitudes de type I* : incertitudes liées à l'environnement et aux conditions d'utilisation, ce sont donc des incertitudes aléatoires. Les variables que nous avons notées V précédemment présentent ce type d'incertitudes. Elles sont indépendantes de la conception du système physique. Elles peuvent être connues soit par des valeurs caractéristiques notées $v_j^{(c)}$ pour chaque variable V_j soit par une densité de probabilité complète obtenue à partir de mesures ;
- *incertitudes de type II* : incertitudes de fabrication des pièces. Les variables géométriques, que nous avons notées U, ne sont connues du concepteur que par leur valeur nominale qu'il peut ajuster dans l'optique d'une conception robuste. Notons u une réalisation de U : à cause des imperfections potentielles du processus de fabrication, les variables U n'ont jamais la valeur nominale souhaitée u^* et une dispersion Δu apparaît autour de u^*

$(u = u^* + \Delta u)$;

- *incertitudes sur les fonctions du système* : incertitudes liées à l'évaluation des performances du système, ce sont des incertitudes épistémiques qui touchent ici la fonction f. On peut parfois les regrouper avec les incertitudes de type I ;
- *incertitudes de faisabilité* : incertitudes sur les fonctions contraintes définissant l'espace de conception. Ce sont encore des incertitudes épistémiques mais qui touchent cette fois les fonctions g_i.

La prise en compte de cet incertain est une condition indispensable pour un résultat optimal et robuste des études menées sur un système. Une fois que les sources d'incertitudes ont été identifiées, il convient de les modéliser et de les quantifier.

1.2 Modélisation des incertitudes

La prise en compte des incertitudes peut se faire de différentes manières :

- *approche au pire des cas (ou déterministe)* : chaque variable incertaine est connue par ses valeurs haute $u_i^{(+)}$ et basse $u_i^{(-)}$ pour les variables de décision $u_i, i = 1, \ldots, d_1$ et une valeur caractéristique $v_j^{(c)}$ pour les variables $V_j, j = 1, \ldots, d_2$;
- *approche probabiliste* : chaque variable incertaine (U et V) est définie par sa densité de probabilité $\tilde{f}_i(u_i), i = 1, \ldots, d_1$ et $\tilde{f}_j(v_j), j = 1, \ldots, d_2$;
- *approche possibiliste* : une mesure floue (cf. (Dubois et al., 2008)) est associée à chaque variable incertaine. Il peut s'agir d'une modélisation déterministe avec une loi sous-jacente inconnue mais dont on connaît une statistique (la moyenne par exemple).

La méthode déterministe est simple mais limitée tandis que la méthode possibiliste est plus générale mais aussi plus complexe. Dans la suite, nous utiliserons l'approche probabiliste, ce qui nous permet de faire un compromis entre la bonne représentation des incertitudes et la simplicité de la méthode.

A chaque variable $X_i, i = 1, \ldots, d$ incertaine est associée une densité de probabilité $\tilde{f}_i(x_i)$. Les variables du système n'étant pas entachées d'incertitudes sont fixées à leur valeur nominale. Elles seront donc considérées comme des paramètres et non plus comme des variables.

Les incertitudes sur les fonctions du système et les incertitudes de faisabilité sont beaucoup plus difficiles à modéliser. On les retrouve souvent dans les codes numériques censés reproduire le comportement physique d'un système. Il existe donc un biais entre les simulations et la réalité. Cependant, l'accès aux résultats dans la réalité n'est pas toujours possible, ce qui rend l'évaluation de ce biais inaccessible en pratique. Avec l'amélioration des systèmes informatiques et des capacités de calculs, nous considérons que les modèles utilisés conduisent à des erreurs

suffisamment faibles pour que leurs incertitudes soient négligées.

La modélisation des incertitudes de types I et II par des lois de probabilité conduit à une variabilité des sorties qui présentent donc également des lois de probabilité. L'obtention d'informations sur ces lois en sortie se fait par la propagation d'incertitudes.

2 Méthodologie de propagation des incertitudes

Partant du modèle physique connu (code de calcul), l'étude d'une variable d'intérêt Y à partir de données stochastiques (vecteur $\mathbf{X}$) permet de déterminer une quantité d'intérêt (moments de la loi de Y par exemple) et donc de réaliser la propagation des incertitudes (cf. Figure III.1) au travers du modèle. Le cheminement inverse consiste à considérer un grand nombre d'évaluations de la quantité d'intérêt afin de hiérarchiser les incertitudes. Cela revient à calculer les sensibilités, au sens probabiliste, c'est-à-dire l'influence des paramètres d'entrée sur la variabilité de la réponse.

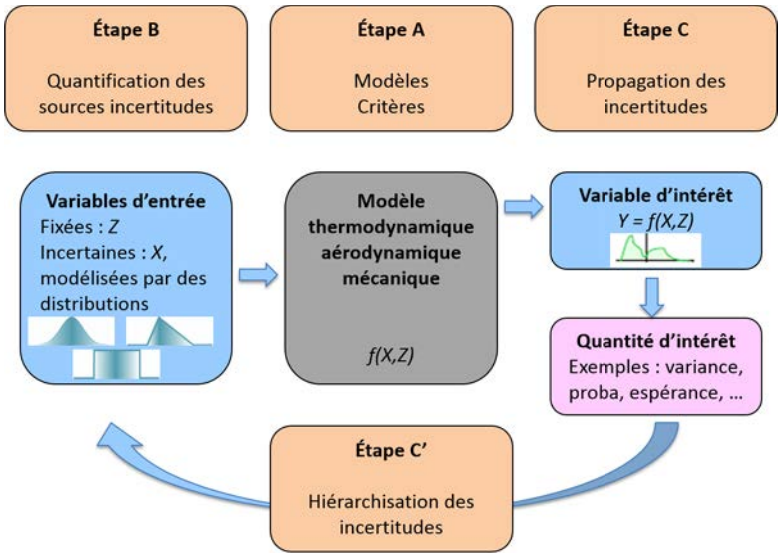

Figure III.1 – Schéma de la "méthodologie incertitude" (de Rocquigny et al., 2008)

Le principal problème de la modélisation probabiliste est que les résultats obtenus en sortie dépendent du choix des différentes lois sur les entrées. Ce phénomène a notamment été étudié par (Lehman et al., 2004). Il faut donc suffisamment d'information sur une incertitude pour choisir la loi qui lui sera attribuée.

Les quantités d'intérêt obtenues en sortie peuvent être obtenues de différentes façons : Monte-Carlo sur le code de calculs, Monte-Carlo sur des modèles locaux, la méthode FOSM (First Order Second Moment) et la méthode URQ (Univariate Reduced Quadrature). Ces quatre méthodes font l'objet des sections suivantes.

2.1 Méthode de Monte-Carlo sur le code de calculs

La méthode la plus évidente est la méthode de Monte-Carlo. Elle consiste à effectuer des tirages de Monte-Carlo à chaque appel du code. Ces tirages sont effectués suivant la loi de chaque variable entachée d'incertitudes et sont répartis autour du point évalué afin de représenter l'incertitude de la variable. La robustesse de la sortie, à travers la (ou les) quantité(s) d'intérêt choisie(s), est donc calculée en chaque point évalué. Les quantités d'intérêt, moments de loi de la sortie Y, sont estimées localement par les moments empiriques de Y sur les N tirages de Monte-Carlo. La moyenne empirique est donnée par (III.1) et l'écart-type empirique par (III.2).

$$m_y = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{III.1}$$

$$\sigma_y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - m_y)^2} \tag{III.2}$$

Du point de vue de l'écart-type, on propage ainsi l'incertitude de la variable X_i sur la sortie Y selon le schéma de la Figure III.2.

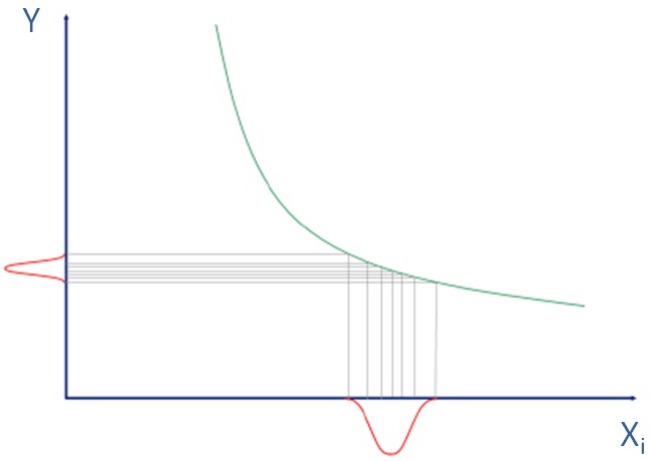

Figure III.2 – Propagation des incertitudes par simulations de Monte-Carlo sur le code de calculs

Chaque simulation selon la loi des entrées fournira une valeur de la sortie. Toutes ces valeurs

permettent d'estimer localement les moments de la loi de cette sortie.

Il est clair que plus le nombre de tirages est important, plus l'estimation de la quantité d'intérêt est précise. Cette méthode peut donc rapidement devenir coûteuse, surtout si le code de calculs lui-même est coûteux. Il peut donc être intéressant d'utiliser un méta-modèle, concept décrit au Chapitre II.

2.2 Méthode de Monte-Carlo sur un modèle local

Cette méthode permet d'estimer les quantités d'intérêt de manière moins coûteuse que les simulations de Monte-Carlo sur le code de calcul. En effet, les simulations de Monte-Carlo effectuées autour du point en cours ne sont plus évaluées via le code de calculs mais par une approximation locale de la fonction étudiée. Une méthode bien connue consiste à considérer une approximation locale linéaire.
L'approximation linéaire de f en un point a consiste à négliger le reste dans le développement limité d'ordre 1 de f au voisinage de a. On écrit alors, pour x dans un voisinage de a

$$f(x) \simeq f(a) + Df(a)(x-a) = f(a) + \sum_{i=1}^{d} \frac{\partial f}{\partial x_i}(a)(x_i - a_i),$$

où $Df(a)$ est la différentielle de f en a qui peut être vu comme le coefficient directeur de l'approximation linéaire. Ceci revient à confondre localement f avec sa tangente. Comme on ne connaît pas forcément l'expression de f, on va se servir de la dérivation numérique pour estimer $Df(a)$.
On choisit $h \in \mathbb{R}^d$ et $(u, v) \in (\mathbb{R}^d)^2$ tels que pour tout $i \in [\![1, d]\!]$, $|u_i - v_i| = h_i$ et $|u_i - a_i| = |v_i - a_i| = h_i/2$. Sur chaque dimension, h_i doit être faible de telle sorte que u et v soient proches. Ainsi, pour $i \in [\![1, d]\!]$, on a

$$\frac{\partial f}{\partial x_i}(a) \simeq \frac{f(a_1, \ldots, a_{i-1}, u_i, a_{i+1}, \ldots, a_d) - f(a_1, \ldots, a_{i-1}, v_i, a_{i+1}, \ldots, a_d)}{h_i}.$$

Sur chaque dimension, ceci revient à remplacer la tangente par la corde. L'estimation de $Df(a)$ nécessite donc d'évaluer $2d$ points.

Les moments de Y sont calculés comme précédemment sauf que les y_i ne sont pas des évaluations du code mais des valeurs approchées par le modèle linéaire. Le schéma de la Figure III.2 devient donc celui de la Figure III.3. Le principal inconvénient de cette méthode est que

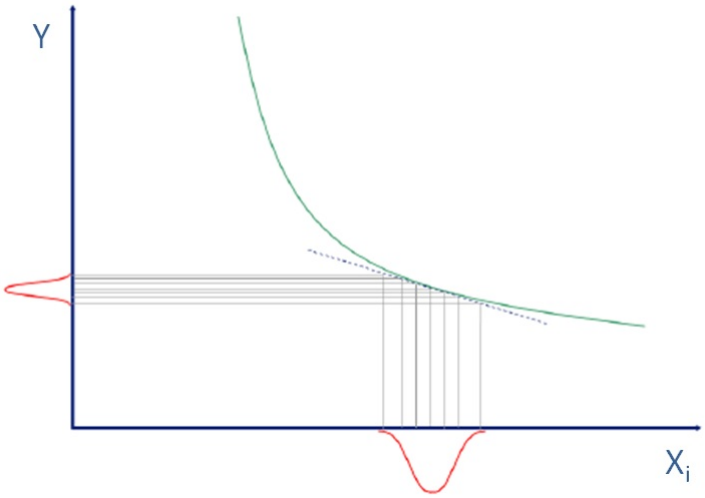

Figure III.3 – Propagation des incertitudes par simulations de Monte-Carlo sur un modèle ajusté localement

la qualité de l'estimation des moments n'est pas bonne si la vraie fonction est discontinue ou présente des non-linéarités importantes.

2.3 Méthode FOSM

La méthode FOSM (First Order Second Moment) est une méthode probabiliste permettant de déterminer les moments stochastiques d'une fonction à partir de variables d'entrée aléatoires. Ainsi, la moyenne de Y sera approchée par (III.3) et l'écart-type par (III.4).

$$m_y \approx f(\mu) \tag{III.3}$$

$$\sigma_y \approx \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial f(\mu)}{\partial x_i}\frac{\partial f(\mu)}{\partial x_j} Cov(X_i, X_j)}, \tag{III.4}$$

où d est la dimension du vecteur d'entrée X et $\mu = (\mu_1, \ldots, \mu_d)$ le vecteur moyenne de X. Ces approximations sont obtenues en remplaçant la fonction f par son développement de Taylor à l'ordre 1.
Notons que si les variables X_i sont indépendantes, c'est-à-dire $Cov(X_i, X_j) = 0, i \neq j$, alors

$$\sigma_y = \sqrt{\sum_{i=1}^{d} \left(\frac{\partial f(\mu)}{\partial x_i}\right)^2 \sigma_{x_i}^2}.$$

Dans le cas d'une seule variable d'entrée ($d = 1$), on a alors que $\sigma_y = \frac{\partial f(\mu)}{\partial x}\sigma_x$. La propagation des incertitudes ressemble à celle de la méthode de Monte-Carlo sur modèle local puisqu'il faut ici aussi estimer les dérivées partielles de f, sauf que nous n'avons plus besoin de simulations de la loi de X pour estimer les moments de la loi de Y (cf. Figure III.4). Cette méthode est plus

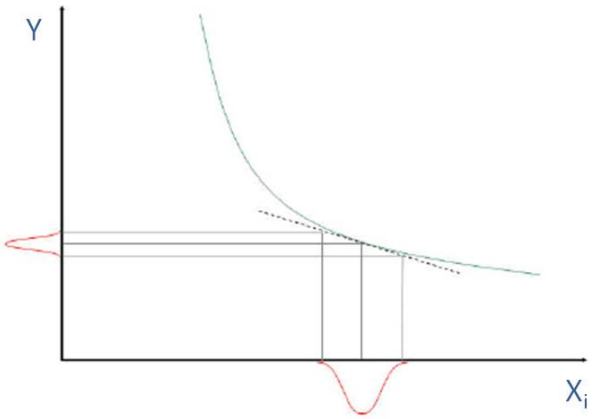

Figure III.4 – Propagation des incertitudes par la méthode FOSM

rapide que la méthode de Monte-Carlo sur code de calculs mais est comparable à la méthode de Monte-Carlo sur modèle local. Comme cette dernière, la méthode FOSM dépend de la forme de la fonction étudiée et peut être en difficulté dans des cas de discontinuités ou de fortes non-linéarités. Le lecteur intéressé par cette méthode pourra se référer à (Haldar and Mahadevan, 2000).

2.4 Méthode URQ

Cette méthode est basée sur la quadrature de Gauss qui permet d'approcher la valeur numérique d'une intégrale à partir de points pris sur le domaine d'intégration. Ces points sont appelés des nœuds. Dans la méthode URQ (Univariate Reduced Quadrature), $2d + 1$ nœuds sont considérés. Le nombre de variables incertaines d dans le système comprend les d_1 entrées et les d_2 paramètres environnementaux, s'il y en a. Elles sont considérées comme indépendantes. Soit $X = (X_1, \ldots, X_d)$ le vecteur des variables aléatoires du système. On note μ_X la moyenne de X et $\sigma_X = (\sigma_{X_1}, \ldots, \sigma_{X_d})$ son écart-type. Les $2d + 1$ points $x = (x^0, x^+, x^-)$ qui seront

évalués sont définis de la manière suivante

$$\begin{aligned} x^0 &= \mu_X \\ x_i^+ &= \mu_X + h_i^+ \sigma_{X_i} e_i, \\ x_i^- &= \mu_X + h_i^- \sigma_{X_i} e_i, \end{aligned}$$

où e_i est le $i^{ème}$ vecteur de la matrice identité de taille d, h_i^+ et h_i^- sont des paramètres d'échelle dépendant des moments centrés réduits d'ordres 3 et 4 (respectivement le coefficient d'asymétrie et le kurtosis notés γ_X et Γ_X) de la loi de X. Ces $2d+1$ points permettent, par la quadrature de Gauss, d'estimer l'espérance et la variance de la sortie $Y = f(X)$ en utilisant les formules suivantes

$$\mu_Y = w_0 f(\mu_X) + \sum_{i=1}^{d} w_i \left(\frac{f(x_i^+}{h_i^+} - \frac{f(x_i^-)}{h_i^-} \right), \tag{III.5}$$

$$\begin{aligned} \sigma_Y^2 = \sum_{i=1}^{d} \Big(& w_i^+ \left(\frac{f(x_i^+) - f(\mu_X)}{h_i^+} \right)^2 + w_i^- \left(\frac{f(x_i^-) - f(\mu_X)}{h_i^-} \right)^2 \\ & + w_i^{\pm} \left(\frac{(f(x_i^+) - f(\mu_X))(f(x_i^-) - f(\mu_X))}{h_i^+ h_i^-} \right) \Big). \end{aligned} \tag{III.6}$$

Par un parallèle entre la quadrature de Gauss et le développement de Taylor de f à l'ordre 4, on obtient les valeurs de h_i^+ et h_i^-

$$h_i^{\pm} = \frac{\gamma_{X_i}}{2} \pm \sqrt{\Gamma_{X_i} - \frac{3\gamma_{X_i}^2}{4}}.$$

Les différents poids présents dans III.5 et III.6 sont alors

$$\begin{aligned} w_0 &= 1 + \sum_{i=1}^{d} \frac{1}{h_i^+ h_i^-}, \\ w_i &= \frac{1}{h_i^+ - h_i^-}, \\ w_i^+ &= \frac{(h_i^+)^2 - h_i^+ h_i^- - 1}{(h_i^+ - h_i^-)^2}, \\ w_i^- &= \frac{(h_i^-)^2 - h_i^+ h_i^- - 1}{(h_i^+ - h_i^-)^2}, \\ w_i^{\pm} &= \frac{2}{(h_i^+ - h_i^-)^2}. \end{aligned}$$

Comparée à la méthode de Monte-Carlo, la méthode URQ permet de réaliser la propagation d'incertitudes plus rapidement et avec moins d'appels au code de calculs. De plus, la précision de l'estimation n'est que légèrement moins bonne que celle fournie par les simulations de Monte-Carlo. Le lecteur intéressé par cette méthode de propagation pourra se référer à (Padulo and Guenov, 2011).
Nous avons défini plusieurs méthodes de propagation des incertitudes à travers un code de calculs. Définissons maintenant le problème de conception robuste afin de déterminer ensuite les méthodes de résolution que nous devrons utiliser pour résoudre un tel problème.

3 Formulation du problème de conception robuste

Les formulations fournies dans cette section sont issues d'un travail de formalisation de la conception robuste effectué à Safran Aircraft Engine.

3.1 Conception admissible

Dans le code de calculs, chaque expérience est déterministe et les variables indépendantes de la conception, V, sont fixées à leur valeur caractéristique $v^{(c)}$. Le but du code de calculs seul est de satisfaire les contraintes du système. Cela consiste donc à trouver des valeurs nominales u_i^* des variables U_i telles que la fonction $g(u^*, v^{(c)}) \geq 0$, où $g = (g_1, ..., g_m)$, soit satisfaite. La formulation mathématique de ce problème est le suivant

$$\text{Trouver } u^*_{Adm} \text{ tel que } g(u^*_{Adm}, v^{(c)}) \geq 0. \tag{III.7}$$

Elle conduit à une conception admissible au nominal. Dans la pratique, pour garantir la meilleure performance possible, le concepteur cherchera à se rapprocher le plus possible de l'égalité.

3.2 Conception déterministe optimale

L'optimisation sans contrainte de la performance du système consiste à rechercher la conception déterministe optimale, ce qui peut être formulé mathématiquement de la façon suivante

$$\text{Trouver } u^*_{Opt} \text{ tel que } u^*_{Opt} = \arg\max_{u^*} f(u^*, v^{(c)}). \tag{III.8}$$

L'analyse de sensibilité déterministe de la fonction f peut aider dans cette démarche en identifiant la direction de recherche privilégiée de ce maximum. Il s'agit d'un problème d'optimisation sans contrainte pouvant conduire à une solution non admissible vis-à-vis des fonctions contraintes.
C'est pourquoi nous considérons plus généralement l'optimisation sous contraintes en ajoutant les contraintes de conception à la formulation précédente.

3.3 Conception optimale et admissible

Ceci conduit à la définition d'une conception optimale vis-à-vis de la fonction objectif et admissible vis-à-vis des fonctions contraintes, ce que l'on formule mathématiquement par

$$\text{Trouver } u^*_{OptAdm} \text{ tel que } \begin{cases} u^*_{OptAdm} = \arg\max_{u^*} f(u^*, v^{(c)}) \\ g(u^*_{OptAdm}, v^{(c)}) \geq 0 \end{cases} . \tag{III.9}$$

Il s'agit d'un problème classique d'optimisation sous contraintes sans incertitude sur les différentes variables.

3.4 Conception robuste

La résolution d'un problème de conception robuste passe par la considération des incertitudes sur les différentes variables en jeu et sur leur impact sur la fonction objectif et non sur les fonctions contraintes (cas de la fiabilité). La robustesse d'un système se mesure par son aptitude à avoir une performance optimale et insensible aux différentes sources d'incertitudes. Il s'agit d'une propriété locale qui consiste à faire un compromis entre moyenne et dispersion de la performance. La forme de la loi de U est fixée, sa dispersion est donc également fixée. Pour chaque valeur nominale u^* évaluée, la loi de U est centrée sur u^*. Il s'agit de la loi conditionnelle de U sachant que $\mathbb{E}[U] = u^*$. On note $U_{u^*}(w)$ la variable aléatoire suivant cette loi conditionnelle et entachée d'incertitudes. On note $U_{u^*}(w_j), j = 1, \ldots, N$ des réalisations de cette variable. De même $V(w_j), j = 1, \ldots, N$ sont des réalisations de la variable V entachée d'incertitudes. La conception robuste est la conception qui rend optimale la robustesse du système. Cette robustesse peut être définie mathématiquement de différentes façons. Vu comme une optimisation bi-objectifs, le problème de conception robuste se présente de la manière suivante

$$\text{Trouver } u^*_{Rob} \text{ tel que } u^*_{Rob} = \arg\min_{u^*} \{\mathcal{E}[f(U_{u^*}(w), V(w))], Var(f(U_{u^*}(w), V(w)))\} \tag{III.10}$$

Au sens de Taguchi, la robustesse est vue comme la maximisation de la fonction de perte, ou *Signal-to-Noise Ratio (SNR)*. Le lecteur intéressé pourra se référer à (Trosset, 1996) où un état des lieux de la méthode Taguchi est réalisé. Lorsque la performance du système n'a pas de valeur cible, la fonction *Mean Square Deviation* (MSD) de Taguchi est définie par une espérance mathématique

$$MSD(u^*) = \mathbb{E}[(f(U_{u^*}(w), V(w)))^2] = \frac{1}{N}\sum_{j=1}^{N}(f(U_{u^*}(w_j), V(w_j)))^2, \tag{III.11}$$

où $f(U_{u^*}(w_j), V(w_j))$ sont des mesures de la performance du système, entachées d'incertitudes. Cet indicateur prend en compte l'espérance et la variance de la performance, dans le sens où

$$\mathbb{E}[(f(U_{u^*}(w), V(w)))^2] = Var(f(U_{u^*}(w), V(w))) + (\mathbb{E}[f(U_{u^*}(w), V(w))])^2 .$$

De plus, il est lié au *Signal-to-Noise Ratio (SNR)* par la relation

$$SNR(u^*) = -10\log_{10}(MSD(u^*)), \tag{III.12}$$

expression plus lisse et donc plus simple à optimiser. La formulation mathématique de la recherche de la conception robuste optimale est donc, au sens de Taguchi,

$$\text{Trouver } u^*_{Rob} \text{ tel que } u^*_{Rob} = \arg\max_{u^*} SNR(u^*), \tag{III.13}$$

ce qui est équivalent à trouver les u^* qui minimise $MSD(u^*)$.

3.5 Conception robuste et admissible

Pour prendre en compte à la fois la robustesse de la conception sans perdre de vue que celle-ci doit être admissible vis-à-vis des fonctions contraintes, on considère la conception robuste et admissible formulée mathématiquement de la façon suivante

$$\text{Trouver } \bar{D}_{RobAdm} \text{ tel que } \begin{cases} u^*_{RobAdm} = \arg\max_{u^*} SNR(u^*) \\ g(u^*_{RobAdm}, v^{(c)}) \geq 0 \end{cases} . \tag{III.14}$$

C'est ce type de problème que nous tenterons de résoudre dans ce chapitre.

3.6 Choix de l'indicateur de robustesse

Nous avons considéré, dans la définition du problème de conception robuste III.13, l'indicateur de robustesse fourni par Taguchi, mais il en existe d'autres. Dans (Beyer and Sendhoff, 2007) est proposée une approche pondérée (*Agregation Approach*) qui consiste à définir un indicateur de la robustesse comme étant

$$I(u^*) = (1-\beta)\mathbb{E}[f(U_{u^*}(w), V(w))] + \beta Var(f(U_{u^*}(w), V(w))), \beta \in [0,1]. \tag{III.15}$$

Le problème de conception robuste s'écrit alors

$$\text{Trouver } u^*_{Rob} \text{ tel que } u^*_{Rob} = \arg\min_{u^*} I(u^*). \tag{III.16}$$

Cet indicateur pondéré permet de considérer simultanément l'espérance et la variance de la fonction objectif, en maximisant la première et en minimisant la seconde. La recherche de la conception robuste, qui est donc un problème d'optimisation multi-objectifs, peut présenter des difficultés puisque le fait de vouloir réduire la variance de f peut être contradictoire avec le fait d'augmenter son espérance. Même lorsque les deux objectifs sont des objectifs de minimisation, il n'est pas rare qu'il s'agisse encore d'objectifs contradictoires. L'approche pondérée permet d'éviter ce problème et de n'effectuer qu'une optimisation mono-objectif. En effet, l'indicateur est une somme pondérée des fonctions objectifs et a pour vocation d'être minimisé. Cependant, si nous n'avons pas d'idée a priori sur le poids des deux objectifs, alors l'optimisation doit être effectuée pour plusieurs valeurs de β.

Pour chaque valeur de β, on trouve une solution Pareto-optimale (Beyer and Schwefel, 2002). Les solutions Pareto-optimales permettent de prendre en compte les contraintes puisqu'elles sont définies par rapport à la région réalisable. La définition ci-dessous est proposée dans (Coello et al., 2002).

Définition 3 (Solution Pareto-optimale)**.** *Soit un problème d'optimisation multi-objectifs ($k \geq 2$ objectifs) sous contraintes. On note $f_1, \ldots, f_k$ les k fonctions objectifs à minimiser ou maximiser. Un point x^* réalisable (satisfaisant toutes les contraintes) est une solution Pareto-optimale pour ce problème si pour tout x réalisable, on a une des deux situations suivantes :*

i. $\forall i \in [\![1,k]\!], f_i(x) = f_i(x^*)$,

ii. $\exists i \in [\![1,k]\!], f_i(x) > f_i(x^*)$ *dans le cas de la minimisation ($f_i(x) < f_i(x^*)$ pour la maximisation).*

Ces points constituent le front de Pareto, but de l'optimisation multi-objectifs. Ces solutions reposent sur le principe qu'il est impossible de trouver une solution meilleure sur un critère sans qu'elle soit plus mauvaise sur l'autre critère. On parle de solutions non dominées. Dans un cas bi-objectifs, le front de Pareto est représenté par une courbe avec en abscisse le premier objectif et en ordonnée le second, comme le montre la Figure III.5 dans le cas de deux minimisations. Dans des cas à $k > 2$ objectifs, le front de Pareto est une surface de dimension $k - 1$. Le front

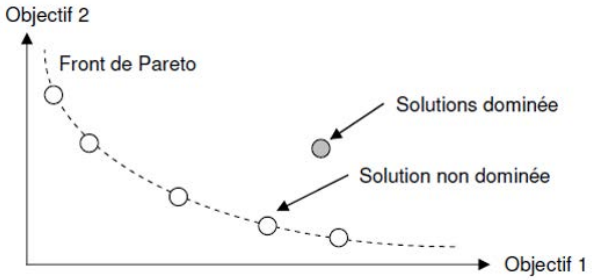

Figure III.5 – Exemple d'un front de Pareto

de Pareto est défini comme étant la frontière entre la région des points faisables (satisfaisant toutes les contraintes) situés au-dessus du front et celle des points infaisables (ne satisfaisant pas au moins une des contraintes), situés en-dessous. Les points blancs appartiennent donc au front de Pareto puisque ce sont des solutions Pareto-optimales. Par contre, le point gris, qui satisfait les contraintes, n'est pas Pareto-optimal puisqu'on peut trouver un point du front qui lui est meilleur vis-à-vis des objectifs.
Dans le cas de la conception robuste, l'espérance de f est en abscisse et la variance en ordonnée. Un problème de conception robuste sera donc vu comme la résolution d'une optimisation multi-objectifs sous contraintes. On a deux objectifs : minimiser ou maximiser l'espérance de f et minimiser son écart-type.

Notons que d'autres quantificateurs de la robustesse sont disponibles dans la littérature, dans le cas où la fonction objectif a une performance cible. Dans ce cas, il s'agit de quantifier la robustesse par la probabilité que la performance cible ne soit pas atteinte, ce qui fournit un indicateur probabiliste de la robustesse établi de la manière suivante

$$I(u^*) = \mathbb{P}(f(U_{u^*}(w), V(w)) \leq s).$$

La recherche de la conception robuste optimale consiste alors à résoudre le problème suivant

$$\text{Trouver } u^*_{Rob} \text{ tel que } u^*_{Rob} = \arg\min_{u^*} I(u^*)$$

Pour résoudre des problèmes d'optimisation multi-objectifs, différentes méthodes existent. Elles sont présentées à la section suivante. Certaines permettent de construire le front de Pareto tandis que d'autres cherchent un unique point solution fournissant un compromis entre les deux objectifs.

4 Méthodes de résolution d'un problème de conception robuste

Le principe d'une optimisation multi-objectifs est différent du principe d'une approche mono-objectif. Le but principal d'une optimisation mono-objectif est de trouver la solution optimale globale qui donne la meilleure valeur (plus petite ou plus grande) de la fonction mono-objectif. Dans un problème d'optimisation multi-objectifs, on dispose de $k \geq 2$ fonctions objectifs, chaque fonction objectif pouvant avoir une solution optimale différente. Le but d'un problème multi-objectifs est de trouver de « bons compromis » plutôt qu'une seule solution. Certaines méthodes permettent de trouver une solution réalisant un compromis entre les différents objectifs. D'autres permettent d'en trouver plusieurs et de construire le front de Pareto. Dans toute la suite, nous noterons f_1 et f_2 les fonctions objectifs correspondant à chacune des deux optimisations que nous supposerons être des minimisations.
La plupart des méthodes décrites ici le sont également dans (Collette and Siarry, 2002).

4.1 Méthodes trouvant un compromis entre les objectifs

Parmi les méthodes effectuant un compromis entre les objectifs de l'optimisation, nous exposerons rapidement quatre méthodes :

- la méthode hiérarchique,
- la méthode par compromis,
- la fonction de distance,
- la méthode du critère global.

Ces méthodes sont étudiées par exemple dans (de Oliveira and Saramago, 2010).

Méthode hiérarchique La méthode hiérarchique consiste à organiser les objectifs par ordre d'importance. Chaque fonction objectif est minimisée individuellement sous une contrainte qui empêche le minimum de la nouvelle fonction d'être supérieur à une fraction choisie du minimum de la fonction objectif précédente.

Dans le cas d'une optimisation bi-objectifs, 1 est le rang de l'objectif le plus important, 2 celui du moins important. Deux optimisations simples à un objectif sont effectuées successivement. La première permet de trouver le point $x^{(1)} = (x_1^{(1)}, \ldots, x_d^{(1)})$ tel que $f_1(x^{(1)}) = \min_x f_1(x)$. La seconde optimisation consiste à trouver $x^{(2)} = (x_1^{(2)}, \ldots, x_d^{(2)})$ tel que

$$f_2(x^{(2)}) = \min_x f_2(x),$$
$$\text{sous la contrainte } f_2(x) \leq (1 + \frac{\epsilon}{100}) f_1(x^{(1)})),$$

où ϵ est un réel positif inférieur à 100. Le choix de ϵ se fait par l'utilisateur. Plus sa valeur est petite, plus la condition est difficile à satisfaire. Le lecteur intéressé par cette méthode pourra se référer à (Osyczka, 1981).

Méthode par compromis Cette méthode consiste à convertir le problème d'optimisation multi-objectifs en un problème d'optimisation simple à un seul objectif. Pour cela, un des deux objectifs est choisi comme étant l'objectif principal tandis que l'autre devient une contrainte du problème. On cherche alors à résoudre

$$\min_x f_1(x) \tag{III.17}$$
$$\text{sous la contrainte } f_2(x) \leq w \min_x f_2(x) \tag{III.18}$$

La quantité $w > 1$ est un poids permettant d'assouplir la contrainte qui serait trop forte avec $w = 1$. Le minimum $\min_x f_2(x)$ est le résultat de l'optimisation simple de f_2. Les fonctions objectifs sont interchangeables, le choix se fait par l'utilisateur, tout comme pour la valeur de w. Le lecteur intéressé par cette méthode pourra se référer à (Nakayama and Sawaragi, 1984).

Fonction de distance Cette méthode est utilisée lorsque les optimisations consistent à atteindre des valeurs cibles a_1 et a_2. La méthode consiste à minimiser l'écart entre les observations de chaque fonction et leur cible respective. Pour cela, on utilise une fonction de distance

$$E_p = (|f_1(x) - a_1|^p + |f_2(x) - a_2|^p)^{1/p}, 1 \leq p \leq \infty,$$

où p est l'ordre de la méthode. Lorsque $p = 1$, on parle de programmation par buts. Pour $p = 2$, il s'agit de la norme euclidienne.

Méthode du critère global Cette méthode consiste à trouver la combinaison des entrées assurant le minimum d'un critère global. Ce critère consiste à établir la proximité de chaque fonction objectif f_i à un minimum a_i^* selon un écart relatif. Le but est de résoudre

$$\min_x \left[\left(\frac{a_1^* - f_1(x)}{a_1^*} \right)^2 + \left(\frac{a_2^* - f_2(x)}{a_2^*} \right)^2 \right].$$

Les minima a_i^* sont les valeurs optimales de chaque fonction objectif f_i du problème d'optimisation simple à un objectif. Le lecteur intéressé par cette méthode pourra se référer à (Rao and Rao, 2009).

Ces quatre méthodes permettent d'obtenir une unique solution au problème d'optimisation multi-objectifs. Les méthodes suivantes consistent à obtenir plusieurs points solutions afin d'établir un front de Pareto.

4.2 Méthodes établissant le front de Pareto

Comme nous l'avons vu précédemment, le front de Pareto est la limite séparant la région des points faisables (satisfaisant toutes les contraintes) des points non-faisables. Dans un cas bi-objectifs, ce front est une ligne continue que l'on représente par rapport aux deux objectifs. Le but des méthodes d'optimisation générant le front de Pareto est de trouver un certain nombre de points sur la ligne de Pareto en donnant différents poids aux différents objectifs. Par exemple, pour obtenir trois points dans un cas bi-objectifs, on pourra considérer les trois jeux de poids : $(1, 0), (0.5, 0.5), (0, 1)$. Nous exposons cinq méthodes de ce type.

Méthode d'intersection de limites normales Cette méthode, appelée *normal-boundary intersection method* en anglais, utilise une paramétrisation géométrique intuitive afin d'établir une répartition équilibrée de points sur le front de Pareto. Ceci permet d'avoir une représentation précise de toute la ligne. Les optima simples sous contraintes de chaque fonction objectif sont déterminés : f_1^* et f_2^*. Ces deux solutions correspondent respectivement aux poids $(1, 0)$ et $(0, 1)$. Elles constituent les extrémités du front de Pareto. La méthode consiste ensuite à trouver les solutions Pareto-optimales sur des lignes perpendiculaires au segment $[f_1^*, f_2^*]$. Les points d'intersections des perpendiculaires à $[f_1^*, f_2^*]$ sont équi-répartis sur le segment. La méthode est représentée à la Figure III.6. Les points bleus ainsi générés sont bien répartis sur le front de Pareto. La zone délimitée par une courbe noire est la région acceptable. Cette méthode a été étudiée dans (Das and Dennis, 1998) puis dans (Shukla, 2007).

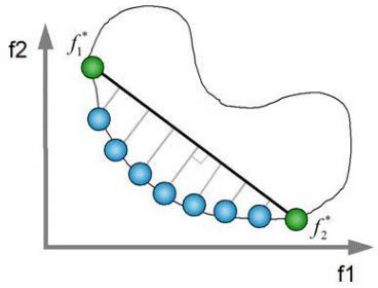

Figure III.6 – Illustration de la méthode d'intersection de limites normales

Méthode à objectifs pondérés La méthode consiste à transformer le problème d'optimisation multi-objectifs en un problème d'optimisation simple à un seul objectif

$$\min_x w_1 f_1(x) + w_2 f_2(x)$$

où $w_i \geq 0$ et $w_1 + w_2 = 1$. Cette optimisation simple est effectuée pour plusieurs choix de poids, le nombre est choisi par l'utilisateur. A chaque jeu de poids correspond une solution de Pareto, comme le montre la Figure III.7. Ici, cinq jeux de poids ont été considérés, ce qui nous donne

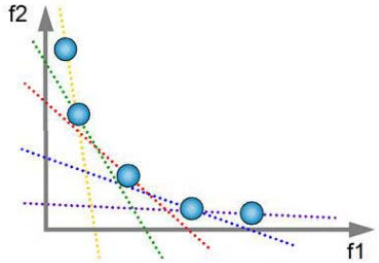

Figure III.7 – Illustration de la méthode à objectifs pondérés

cinq points du front de Pareto. Le lecteur intéressé par cette méthode pourra se référer à (Kim and Weck, 2005) et (Marler and Arora, 2010).

Méthode de Tchebycheff pondérée On considère les valeurs minimales de f_1 et f_2 obtenues par optimisation simple notées f_1^* et f_2^*. La méthode de Tchebycheff pondérée consiste à résoudre le problème

$$\min_x \max(w_1|f_1(x) - f_1^*|, w_2|f_2(x) - f_2^*|),$$

avec $w_1 + w_2 = 1$. Les poids prennent différentes valeurs afin de trouver différents points sur le front de Pareto. Cette méthode a été étudiée dans (Steuer and Choo, 1983) et (Miettinen, 2001).

Méthode de l'optimum min-max Proche de la méthode précédente, la méthode de l'optimum min-max consiste à résoudre le problème

$$\min_x \max \left(w_1 \frac{f_1(x) - f_1^*}{f_1^*}, w_2 \frac{f_2(x) - f_2^*}{f_2^*} \right),$$

où f_1^* et f_2^* sont déterminés comme précédemment et $w_1 + w_2 = 1$. Différents jeux de poids fournissent différents points sur le front de Pareto. Le lecteur intéressé pourra se référer à (Zhang and Gao, 2006).

Méthode génétique de tri basée sur la non-dominance Cette méthode, appelée *non-dominated sorting genetic method*, s'appuie sur les méthodes génétiques dont l'application dans les cas mono-objectif est décrit au Chapitre IV. Dans un premier temps, définissons la dominance dans le cas de deux minimisations :

Définition 4 (Dominance)**.** *Une solution A domine une solution B si et seulement si :* $\forall i \in \{1,2\}, f_i(A) \leq f_i(B)$ *et* $\exists j \in \{1,2\}, f_j(A) < f_j(B)$.

Si la solution A domine la solution B, on dit que B est dominée par A ou bien A est non dominée par B ou entre les deux solutions, A est la solution non dominée.

L'algorithme associé à la méthode étudiée ici, noté NSGA-II et proposé dans (Deb et al., 2002), a les caractéristiques suivantes :

- il utilise une approche élitiste pour sauvegarder les meilleures solutions trouvées lors des itérations précédentes,
- il utilise une procédure rapide de tri basée sur la non-dominance,
- il utilise un opérateur de comparaison basé sur une distance de surpeuplement,
- il n'a aucun paramètre à régler.

Notons que la distance de surpeuplement est une mesure de proximité d'un point avec ses voisins. Une grande distance moyenne de surpeuplement sur un échantillon signifie qu'il y a une grande diversité dans la population étudiée. A chaque étape de l'algorithme, on dispose d'une population (R_t) de $2N$ individus. Elle est séparée en une population de parents (P_t) de taille N et une population d'enfants (Q_t) de taille N. La population de taille $2N$ est ensuite triée selon un critère de non-dominance pour identifier différents fronts $F_1, F_2, \ldots$. Le premier front est l'ensemble des individus non-dominés, le deuxième comprend les individus dominés uniquement par ceux du premier front, et ainsi de suite. Les meilleurs individus vont donc se retrouver dans le ou les premiers fronts. Une nouvelle population parent (P_{t+1}) est formée

en considérant les meilleurs individus appartenant aux premiers fronts. Tous les individus de chaque front successif sont sélectionnés tant que le nombre d'individus ne dépasse pas N. Si le nombre d'individus présents dans (P_{t+1}) est inférieur à N, on va sélectionner les individus du front suivant, qui n'a pas été inclus dans (P_{t+1}), et appliquer une procédure de surpeuplement pour atteindre les N individus. Une fois que les individus appartenant à la population (P_{t+1}) sont identifiés, une nouvelle population enfant (Q_{t+1}) est créée par sélection, croisement et mutation.

La sélection est basée sur un opérateur de comparaison. Cet opérateur permet d'ordonner deux solutions selon leur rang et la valeur de la distance de surpeuplement. Entre deux solutions de rangs différents, on préfère la solution avec le plus petit rang (ou le plus petit front). Pour deux solutions qui appartiennent au même front, on préfère la solution qui est localisée dans la région où la densité de solutions est moindre, soit l'individu possédant la plus grande valeur de distance de surpeuplement.

Le fonctionnement de l'algorithme est représenté à la Figure III.8.

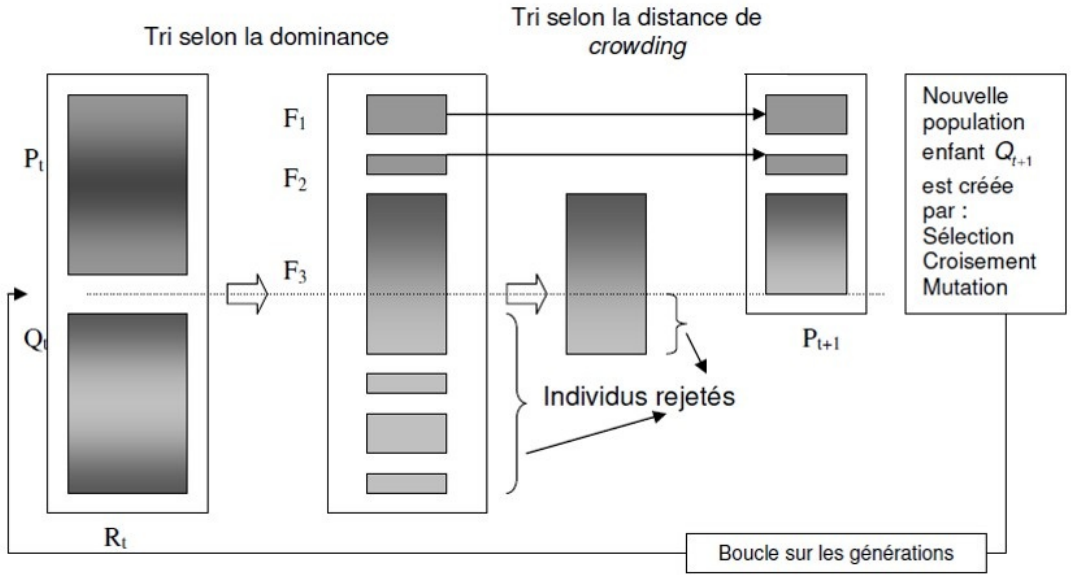

Figure III.8 – Illustration de la méthode génétique de tri basée sur la non-dominance

5 Optimisation robuste de la masse sous contraintes

La mise en place d'une méthode de conception robuste présente trois intérêts majeurs :

- la réduction des temps de conception,
- la robustesse des résultats obtenus,

- l'augmentation de la qualité du produit.

Le problème consiste à minimiser de manière robuste la masse résultant du dimensionnement aéro-dynamique, tout en satisfaisant les 10 contraintes du système. Nous faisons varier les trois entrées A, C et G sélectionnées au Chapitre II comme étant les plus influentes sur la masse. On note f le code de calculs permettant d'obtenir la masse et $g_i, 1 \leq i \leq 10$ les fonctions représentant les contraintes. Le problème à résoudre est le suivant

$$\begin{cases} \min\limits_{x \in D} S(x) = \{\mathbb{E}[f(x)], Var(f(x))\} \\ \text{tel que } g_i(x) \geq 0, 1 \leq i \leq 10. \end{cases} \tag{III.19}$$

5.1 Types d'incertitudes en avant-projets

Le processus de dimensionnement peut être représenté comme trois boîtes successives. Chaque boîte comporte ses propres incertitudes plus le résultat des incertitudes des boîtes précédentes. Ceci est illustré à la Figure III.9 qui reprend le processus avant-projets décrit dans la Section 1.1 du Chapitre I.

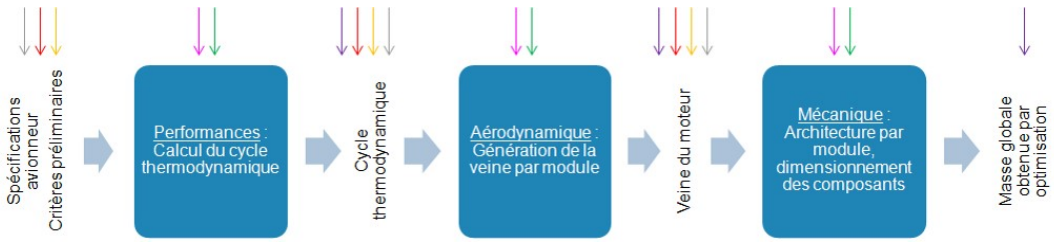

Figure III.9 – Les différentes sources d'incertitudes en avant-projets

Chaque couleur de flèche correspond à un type d'incertitudes :

- → erreurs d'estimation,
- → erreurs de prévision,
- → incertitudes sur les données d'entrée,
- → incertitudes liées à la forme des modèles,
- → incertitudes de prédiction du modèle,
- → incertitudes liées à l'outil.

Dans le processus décrit à la Figure III.9, chaque étape présente des incertitudes de types I et II sur les entrées (estimation, fixation de variables, hypothèses simplificatrices), des incertitudes sur les sorties du type incertitudes sur les fonctions et incertitudes de faisabilité (erreurs de prédiction des modèles). Ces incertitudes touchent également les boîtes elles-mêmes (incertitudes liées à l'outil et à la forme des modèles). Seuls les experts de chaque métier ou les retours

d'expériences permettent d'identifier et de quantifier ces incertitudes.
Le problème en avant-projets est que, comme son nom l'indique, nous nous situons avant la conception physique du moteur. Cette phase ne dispose donc pas de mesures physiques et les retours d'expériences ne sont disponibles que pour les moteurs antérieurs. En effet, les codes de calculs utilisés durant cette phase sont déterministes, donc incapables de quantifier l'influence de perturbations sur les performances du système. La conception robuste permet d'intégrer dans les codes de calculs les incertitudes et les risques liés aux perturbations.
Pour cela, nous considérerons des lois normales sur les entrées, ce qui est courant dans les dimensionnements. Il n'est pas rare de les tronquer afin d'assurer que les valeurs prises ne soient pas trop éloignées de la moyenne. Par contre, nous ne disposons pas de paramètres environnementaux. De plus, il nous est difficile de quantifier les incertitudes liées aux outils et aux modèles utilisés. Cela nécessiterait une étude comparative entre les résultats fournis par le code de calculs et les dimensions réelles de la pièce une fois conçue, ce dont nous ne disposons pas. Les seules incertitudes que nous prendrons en compte ici sont les incertitudes de type II sur les entrées du système. Une fois toutes ces incertitudes recensées et quantifiées, il faut étudier leur propagation à travers tout le processus de dimensionnement.

5.2 Stratégie de propagation des incertitudes

Parmi les trois méthodes de propagation des incertitudes décrites à la section 2, nous choisissons d'utiliser les simulations de Monte-Carlo évaluées directement sur le code de calculs. Nous avons vu au Chapitre II que la masse ne pouvait être modélisée par une fonction linéaire. Elle présente même de fortes non-linéarités. Utiliser la méthode de Monte-Carlo sur un modèle local linéaire pour propager les incertitudes ne semble donc pas adapté ici. D'autre part, la méthode FOSM exige le calcul de dérivées partielles de la fonction de masse, qui ne sont pas forcément calculables dans notre cas. Nous aurions pu utiliser la méthode URQ mais elle n'est pas implémentée dans Optimus que nous utilisons pour effectuer nos calculs.
Comme il faut environ 2 minutes pour effectuer un calcul, nous ne pouvons pas considérer trop de simulations de Monte-Carlo. En effet, un choix de 10 simulations pour une optimisation nécessitant 1000 appels à la fonction conduit à 10^4 évaluations, ce qui peut prendre jusqu'à 6 heures de calculs.

		NBIM	Min-max	NSGA
Poids $(1, 0)$	$\bar{m}$	67.22	63.84	59.63
	$\sigma(m)$	2.8	8.78	3.26
	Contraintes	NOK	OK	NOK
Poids $(0.75, 0.25)$	$\bar{m}$	69.14	63.63	-
	$\sigma(m)$	2.94	6.75	-
	Contraintes	OK	NOK	-
Poids $(0.5, 0.5)$	$\bar{m}$	69.12	63.88	59.69
	$\sigma(m)$	4.34	7.14	3.1
	Contraintes	OK	OK	NOK
Poids $(0.25, 0.75)$	$\bar{m}$	69.11	91.53	-
	$\sigma(m)$	5.43	29.62	-
	Contraintes	NOK	NOK	-
Poids $(0, 1)$	$\bar{m}$	69.29	75.39	62.09
	$\sigma(m)$	7.06	21.2	2.02
	Contraintes	OK	OK	NOK
	Temps de calculs	4 jours 16 heures	3 jours 20 heures	2 jours 20 heures

Tableau III.1 – Test des méthodes d'optimisation multi-objectifs pour la conception robuste de la masse

5.3 Application des méthodes d'optimisation multi-objectifs pour la conception robuste

Nous cherchons un optimum robuste admissible en appliquant les méthodes fournissant des fronts de Pareto. Les méthodes appliquées, utilisées dans Optimus, sont les suivantes

- méthode d'intersection de limites normales (NBIM),
- méthode de l'optimum min-max,
- méthode génétique de tri basée sur la non-dominance (NSGA).

Selon les méthodes, on obtient des résultats différents. Ces résultats sont répertoriés dans le Tableau III.1. Pour chaque jeu de poids, on obtient la valeur moyenne de la masse $\bar{m}$ et l'écart-type de la masse $\sigma(m)$. La masse est exprimée en kilogrammes. Dans la colonne « C », « OK » signifie que toutes les contraintes sont satisfaites, « NOK » signifie qu'au moins une contrainte n'est pas satisfaite. Le fait que la méthode ne trouve pas l'optimum admissible peut venir du fait que l'algorithme a été arrêté (nombre maximum de points atteint) avant d'avoir convergé. Le fait que le problème soit très contraint (moins de 20% des points satisfont toutes les contraintes) rend l'optimisation plus difficile et peut conduire à des situations où l'optimum admissible n'existe pas pour certains jeux de poids.

Nous remarquons de plus que toutes les méthodes sont très coûteuses en temps de calculs. En pratique, de tels temps peuvent s'avérer rédhibitoires.

La méthode à objectifs pondérés a également été testée pour trois jeux de poids. En 3 jours et 3 heures, on obtient 3 optima. Aucun ne satisfait toutes les contraintes.
Le meilleur optimum est celui de la méthode NSGA. Toutes les contraintes sont satisfaites sauf le clash entre le palier 3 et la veine, qui est la contrainte la plus difficile à satisfaire. Les autres optima ont soit une masse moyenne trop élevée soit un écart-type trop important. On note $(\bar{m}_{opt}, \sigma(m)_{opt})$ le meilleur optimum retenu.
On tentera, grâce aux méthodes d'inversion décrites dans le chapitre suivant, de résoudre ce problème d'intégration (un clash n'est pas positif) tout en assurant une bonne masse. Pour cela, on ajoute une contrainte portant sur la masse. On peut choisir par exemple de considérer qu'avec notre optimum, on a l'intervalle de confiance $[\bar{m}_{opt} - 2\sigma(m)_{opt}, \bar{m}_{opt} + 2\sigma(m)_{opt}]$. Si la variable masse suit une loi normale, alors cet intervalle est un intervalle de confiance à 95%. Nous choisissons donc la contrainte suivante : $masse \leq \bar{m}_{opt} + 2 \times \sigma(m)_{opt}$.

6 Conclusion

Nous avons fait le choix d'une formulation pour la conception robuste parmi d'autres issues de la littérature et de la pratique. Une fois le problème établi, il faut recenser et quantifier les différentes sources d'incertitudes. Ces incertitudes peuvent être de différents types et touchent différentes composantes du système. Ces incertitudes se propagent jusqu'aux sorties par le code de calculs. Pour cela, il faut choisir une méthode de propagation permettant d'estimer les moments de la sortie d'intérêt. Enfin, la conception robuste peut être menée par une méthode d'optimisation bi-objectifs choisie. Il en existe plusieurs dont les résultats peuvent être différents. En effet, certaines consistent à faire un compromis entre les deux objectifs. D'autres conduisent à la construction du front de Pareto en déterminant plusieurs solutions par différentes pondérations des objectifs.
Appliquée au cas de l'optimisation robuste de la masse du CHP, cette méthodologie fournit des résultats qui peuvent être différents selon la méthode choisie. De plus, on remarque que les temps de calculs sont très importants, ce qui peut s'avérer rédhibitoire pour une application industrielle. Selon le temps d'exécution d'un calcul, il peut être judicieux de préférer une méthode effectuant un compromis entre les deux objectifs. Cela permet de ne réaliser qu'une optimisation au lieu de plusieurs pour établir le front de Pareto.
Parmi les tests que nous avons effectués, nous avons choisi le meilleur optimum, c'est-à-dire celui qui assure le meilleur compromis entre la minimisation de la masse et la minimisation de son écart-type qui ne doit pas représenter un pourcentage trop important de la moyenne pour

que l'optimum soit considéré comme robuste. L'optimum obtenu est bon mais ne satisfait pas la contrainte du clash entre le palier 3 et la veine. C'est pourquoi nous faisons appel aux méthodes d'inversion, décrites dans le Chapitre IV. Le but est de résoudre ce problème d'intégration tout en garantissant la proximité à la masse optimale qui devient une contrainte supplémentaire.

Chapitre IV

Résolution de problèmes inverses mal posés

Ce chapitre est consacré à la résolution de problèmes inverses. Ces problèmes consistent à trouver les valeurs des entrées qui assurent une valeur cible de la sortie. Ils ont largement été étudiés dans la littérature, sous différentes formes et différentes appellations. Certaines méthodes bien connues seront décrites dans la deuxième section de ce chapitre. La plupart de ces méthodes permettent de résoudre des problèmes dits bien posés, une notion qui a été définie par Hadamard et qui sera exposée dans la première section. Cependant, il est rare que les problèmes inverses soient des problèmes bien posés, on dit alors qu'ils sont mal posés. Ce sont les problèmes qui nous intéresserons ici et que nous tenterons de résoudre.
Les deux sections suivantes seront respectivement consacrées à deux nouvelles méthodes qui ont été développées durant la thèse. Il s'agit de la méthode MRM et de la méthode COMET.
Les deux dernières sections consisteront à appliquer les méthodes MRM et COMET aux cas tests industriels. Dans un premier temps, nous résoudrons un problème d'intégration sur le cas test principal en utilisant les deux méthodes développées. Puis nous réaliserons l'évaluation de l'effort à partir de signaux de jauges sur le cas test secondaire, où nous verrons que les méthodes développées ne sont pas les plus adaptées puisqu'il est suffisant d'utiliser les méthodes usuelles.

1 Principe des problèmes inverses mal posés

Les problèmes inverses sont explicitement représentés par le schéma de la Figure IV.1. Pour des entrées x, une (ou plusieurs) sortie(s) y et une fonction f, un problème inverse consiste à trouver x pour une valeur de y, connaissant f. En ce sens, il s'agit bien de l'inverse du problème

direct qui permet de trouver y, connaissant x et f. Avec les notations du Chapitre I, on peut

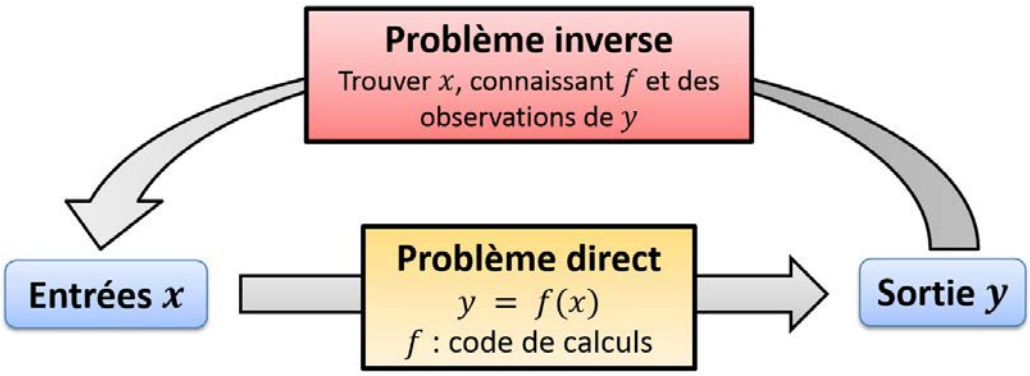

Figure IV.1 – Définition d'un problème inverse

exprimer le problème inverse très simplement comme la résolution de l'équation

$$f(x) = a \tag{IV.1}$$

où a, un réel fixé, est la valeur cible dont la sortie y peut prendre la valeur et $x \in \mathbb{R}^d$, $d \geq 1$ est le vecteur des entrées. En général, les problèmes inverses sont des problèmes mal posés, d'autant plus lorsque la fonction étudiée est une fonction multivariée à valeurs réelles, $f : \mathbb{R}^d \to \mathbb{R}$.
Pour comprendre ce qui caractérise un problème mal posé, définissons tout d'abord ce qu'est un problème bien posé (Kirsch, 2011).

Definition 1 (Problème bien posé)**.** *Soient X et Y deux espaces normés, $f : X \to Y$ une application. L'équation $f(x) = y$ est dite bien posée si :*

i. Existence : pour tout y, il existe au moins un x tel que $f(x) = y$ (surjectivité de f).
ii. Unicité : pour tout y, il y a au plus un x tel que $f(x) = y$ (injectivité de f).
iii. Stabilité : la solution x dépend continûment de y (continuité de f^{-1}).

Les équations pour lesquelles au moins une de ces propriétés n'est pas satisfaite sont dites mal posées.

Ce type de problème peut être trouvé dans la littérature sous des appellations différentes. On parle par exemple de recherche d'antécédents ou d'ensembles de niveaux. En effet, on cherche l'ensemble $S = \{x \in \mathbb{R}^d, f(x) = a\}$ qui représente tous les antécédents de a par f qui peuvent également être vus comme les points de $\mathbb{R}^d$ de niveau $y = a$.
Dans le but de simplifier les notations, le problème est ramené, sans perte de généralité, à la recherche des zéros de f, en considérant $a = 0$. On cherche donc à déterminer le noyau de f : $Ker(f) = \{x \in D : f(x) = 0\}$, dont nous gardons la notation S. L'ensemble $D = [-1, 1]^d, d > 1$

est le domaine de variation normalisé des entrées. La normalisation des entrées est expliquée à la section 1 du Chapitre II. Si f a une forme polynomiale, on parle également de la recherche des racines de f.

2 État de l'art des méthodes de résolution de problèmes inverses

Résoudre des problèmes inverses dans le cas mal posé est l'objectif de ce chapitre. Pour cela, il existe plusieurs méthodes :

- des méthodes usuelles applicables après une étape de régularisation,
- des méthodes d'optimisation,
- une méthode déterministe par grilles,
- une méthode probabiliste,
- une méthode stochastique,
- deux nouvelles méthodes que nous proposons dans cette thèse.

Les trois premiers points seront exposés dans cette section, les méthodes développées au cours de cette thèse seront l'objet des deux sections suivantes.

2.1 Méthodes classiques après régularisation

Si le problème mal posé est ramené à un problème bien posé, alors on peut appliquer des méthodes usuelles de recherche de zéro ou d'optimisation. Pour cela, il faut passer par une étape de régularisation qui peut être plus ou moins simple suivant les cas.

Définition 5 (Image de f)**.** *Soit $f : D \to \mathbb{R}$, avec $D \subset \mathbb{R}^d, d > 1$. L'ensemble image de f est*

$$Imf = \{y \in \mathbb{R} : \exists x \in D, f(x) = y\} = f(D). \tag{IV.2}$$

2.1.1 Méthodes de régularisation

La régularisation d'un problème mal posé consiste à introduire des informations a priori sur le système, souvent sous forme de pénalités. Ces pénalités sont différentes selon la propriété du problème bien posé qui est visée (existence, unicité ou stabilité). Pour les deux premières propriétés, il est souvent assez simple de régulariser le problème.
Le problème de non-existence de la solution peut être réglé en ajustant les bornes de l'espace

d'entrée. En effet, l'existence de la solution à l'équation IV.1 est assurée si $a \in Imf$. La non-existence peut également être résolue en reformulant le problème, souvent sous la forme d'une optimisation par moindres carrés

$$\min_{x \in D} |f(x) - a|^2. \tag{IV.3}$$

Le problème de non-unicité peut également être réglé en ajustant les bornes de variations de x, mais aussi en exigeant de trouver la solution la plus proche d'un point choisi (par exemple la solution de norme minimale) ou en optimisant un critère fixé. Les deux dernières techniques reposent sur l'ajout de contraintes au problème étudié, exprimé par exemple sous forme de pénalisation dans le problème des moindres carrés.
La propriété la plus difficile à assurer est la stabilité (continuité de la solution vis-à-vis de y). Les deux méthodes de régularisation les plus utilisées dans le cas où f est linéaire sont la méthode de Tikhonov (Tikhonov et al., 2013) et la troncature spectrale. Ces méthodes permettent plus généralement de régulariser le problème (IV.3) dans le cas des opérateurs linéaires (Morozov, 2012).
La méthode de Tikhonov s'applique également pour des cas non-linéaires (Engl et al., 1996). Le principe général consiste à considérer un a priori $x_0 \in D$ tel que $f(x_0) = 0$. Le problème (IV.3) est transformé, pour un paramètre de régularisation λ par

$$\min_{x \in D} \{|f(x) - y|^2 - \lambda^2 ||x - x_0||\}. \tag{IV.4}$$

Dans un cas linéaire, il a été prouvé que la solution de (IV.4) avait une solution unique dépendant continûment de y. Il a également été démontré que la méthode convergeait vers la solution la plus proche de x_0. Le choix de λ, comme dans toute méthode de pénalisation, est très important. Ici, s'il est choisi trop petit alors le problème restera mal posé, s'il est choisi trop grand alors la solution sera forcée à se trouver trop près de x_0.

Les méthodes de régularisation nécessitent souvent une bonne connaissance du système étudié, ce qui n'est pas toujours le cas en pratique.

2.1.2 Méthodes usuelles de recherche de zéro

Si le problème est bien posé ou s'il a pu être régularisé, il est possible d'utiliser des méthodes de recherches de zéros, basées sur des algorithmes numériques, permettant de trouver le réel α annulant une fonction $f : f(\alpha) = 0$. Les méthodes les plus connues pour la recherche des zéros d'une fonction sont celles basées sur l'algorithme de Newton-Raphson. Cet algorithme permet

de trouver numériquement une approximation précise d'un zéro d'une fonction différentiable, réelle et à valeurs réelles. Les algorithmes les plus utilisés sont :

- dichotomie,
- point fixe,
- sécante,
- fausse position,
- Müller,
- interpolation quadratique inverse,
- Brent,
- gradient conjugué.

Certaines de ces méthodes ont notamment été étudiées dans (Süli and Mayers, 2003). L'existence d'une solution à l'équation $f(x) = 0$ est assurée par le théorème des valeurs intermédiaires.

Théorème 1 (Théorème des valeurs intermédiaires)**.** *Si f est une fonction continue sur $[a, b]$ et si $f(a)f(b) \leq 0$, alors il existe au moins un point $c \in [a, b]$ tel que $f(c) = 0$.*
Si de plus f est strictement monotone sur $[a, b]$, alors le zéro (ou l'antécédent de 0) est unique dans $[a, b]$.

Méthode de dichotomie La méthode de dichotomie, appelée aussi méthode de bissection, est l'algorithme le plus simple pour trouver les zéros d'une fonction f qui doit satisfaire un certain nombre d'hypothèses, à savoir :

1. le zéro recherché est localisé dans un intervalle donné $[a, b]$;
2. la fonction f doit être continue sur cet intervalle ;
3. il n'y a qu'un seul zéro dans $]a, b[$;
4. les images des bornes de l'intervalle doivent être de signes contraires, c'est-à-dire $f(a)f(b) < 0$. Cette hypothèse assure, par le théorème des valeurs intermédiaires, l'existence d'un réel $x \in]a, b[$ tel que $f(x) = 0$.

Le principe est le suivant. On pose d'abord $c = \frac{a+b}{2}$, le milieu de l'intervalle $[a, b]$. Deux cas de figure sont alors possibles :

1. Si $f(c) = 0$, alors c est la solution et l'algorithme s'arrête.
2. Si $f(c) \neq 0$, il faut regarder les signes de $f(a)f(c)$ et $f(c)f(b)$
   (a) Si $f(a)f(c) < 0$, alors $\alpha \in]a, c[$
   (b) Si $f(c)f(b) < 0$, alors $\alpha \in]c, b[$

On recommence le processus en prenant l'intervalle $[a, c]$ dans le cas (a), l'intervalle $[c, b]$ dans le cas (b). On construit ainsi par récurrence trois suites (a_n), (b_n) et (c_n) telles que $a_0 = a$,

$b_0 = b$ et telles que pour tout $n \geq 0$,

1. $c_n = \frac{a_n + b_n}{2}$.
2. Si $f(c_n)f(b_n) < 0$, alors $a_{n+1} = c_n$ et $b_{n+1} = b_n$.
3. Si $f(c_n)f(a_n) < 0$, alors $a_{n+1} = a_n$ et $b_{n+1} = c_n$.

Le critère d'arrêt de cet algorithme porte sur la proximité de c_n à la solution : on s'arrête dès que, pour une précision $\epsilon > 0$ fixée, $|f(c_n)| \leq \epsilon$.
Cette méthode simple comporte un certain nombre d'avantages et d'inconvénients. En effet, elle s'adapte au cas où f est un code de calcul, on ne connaît donc pas explicitement son expression. De plus, sous les hypothèses 2 et 4, la méthode converge toujours, avec une vitesse de convergence linéaire, ce qui est plutôt lent. Une des particularités de cet algorithme est qu'il est possible de connaître à l'avance le nombre d'itérations nécessaires pour déterminer le zéro de la fonction avec la précision souhaitée : $n \geq \frac{log(\frac{b-a}{\epsilon})}{log(2)} - 1$.

Méthode de point fixe Le principe de cette méthode consiste à transformer l'équation $f(x) = 0$ en une équation équivalente $g(x) = x$ où g est une fonction auxiliaire bien choisie. Le point α, zéro de f, sera un point fixe de g. Le choix de la fonction g est motivé par les exigences du théorème du point fixe.
La seule hypothèse est que la fonction g doit être contractante dans un voisinage I de α, ce qui revient à vérifier que $|g'(x)| < 1$ sur ce voisinage.
Dans ce cas, on construit une suite $(x_n)_{n\in\mathbb{N}}$ définie par :

$$\begin{cases} x_0 \text{ dans un voisinage } I \text{ de } \alpha \\ \forall n \geq 0, x_{n+1} = g(x_n). \end{cases}$$

On applique alors, localement, le théorème du point fixe de Picard. Le critère d'arrêt est atteint pour une précision ϵ dès que $|x_{n+1} - x_n| < \epsilon$.

Le point α est un point fixe attractif de g donc la suite (x_n) converge vers α. L'ordre de convergence de (x_n) est égal à m si et seulement si g est de classe C^m, $g'(\alpha) = \ldots = g^{m-1}(\alpha) = 0$ et $g^{(m)}(\alpha) \neq 0$. Ces conditions exigent une bonne connaissance de la fonction f.
En général, on considère $g(x) = x + af(x)$ où a est un réel non nul donné.

Méthode de Newton-Raphson Le principe de la méthode est qu'à chaque itération i, la fonction f est linéarisée en le point courant x_i et x_{i+1} est le zéro de la fonction linéarisée. Un certain nombre d'hypothèses sont donc nécessaires :

1. la fonction f doit être différentiable aux points visités pour pouvoir y linéariser la fonction ;
2. les dérivées ne doivent pas s'y annuler, pour que la fonction linéarisée ait un zéro ;
3. le premier point x_0 doit être pris suffisamment proche d'un zéro régulier de f (c'est-à-dire en lequel la dérivée de f ne s'annule pas) pour assurer la convergence du processus.

La méthode de Newton-Raphson peut être vue comme une méthode particulière de point fixe. Ici, la fonction g est définie de la manière suivante : $g(x) = x - \frac{f(x)}{f'(x)}$. La méthode itérative s'écrit donc de la manière suivante

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{IV.5}$$

L'algorithme est illustré à la Figure IV.2. La fonction f est représentée par la courbe rouge, elle coupe l'axe des abscisses en α ($f(\alpha) = 0$). On part de x_0, le point d'initialisation choisi. On trace la tangente de f en ce point. L'intersection entre cette tangente et l'axe des abscisses donne x_1. L'obtention de x_2 se fait en traçant la tangente de f en x_1 et en prenant son intersection avec l'axe des abscisses. Et ainsi de suite jusqu'au critère d'arrêt. L'algorithme s'arrête lorsqu'on a

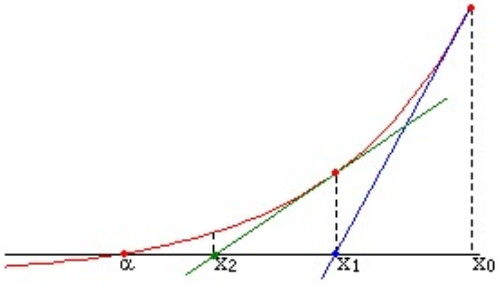

Figure IV.2 – Illustration de l'évolution de l'algorithme de Newton

atteint la précision $\epsilon > 0$ voulue : $|f(x_n)| < \epsilon$. Cette méthode converge à une vitesse quadratique et s'interprète géométriquement comme une méthode de la tangente.
Son principal défaut est qu'il faut calculer explicitement la dérivée de la fonction, donc connaître suffisamment cette dernière. Cette dérivée ne doit pas s'annuler ce qui implique que f ne doit pas avoir des zéros multiples. Elle a par contre l'avantage d'être généralisable à la résolution numérique des équations non linéaires. Elle peut résoudre les systèmes à n équations et n inconnues que l'on ramène à la recherche des zéros d'une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^n$ qui devra être différentiable.

Méthode de la sécante La méthode de la sécante, appelée aussi méthode de Lagrange, est une variante de la méthode de Newton-Raphson où, à chaque itération, la dérivée $f'(x_n)$

est estimée par la pente entre deux points de la fonction

$$f'(x_n) \simeq \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}.$$

Cette estimation fait que l'algorithme perd en efficacité par rapport à la méthode de Newton-Raphson, sa vitesse de convergence est lente. L'itération principale de l'algorithme est

$$x_{n+1} = x_n - \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}. \tag{IV.6}$$

La méthode nécessite donc deux points de départ. Elle est illustrée à la Figure IV.3. La fonction f, en bleu, coupe l'axe des abscisses au point vert. Les deux points de départ sont les points

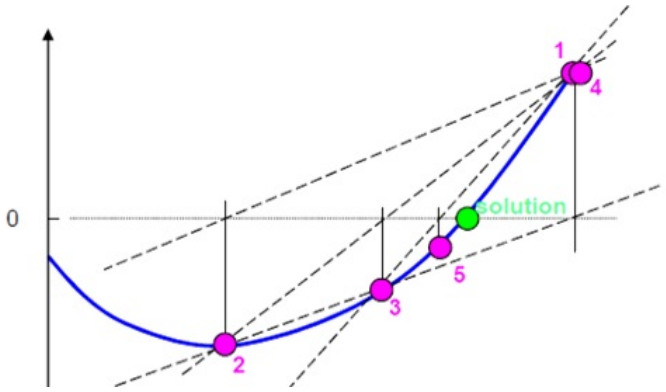

Figure IV.3 – Illustration de l'évolution de l'algorithme de la sécante

1 et 2. On trace la corde entre ces deux points, correspondant à la pente. L'intersection entre ce segment et l'axe des abscisses donne le point 3. On trace alors la corde entre les points 2 et 3, qui, intersectée avec l'axe des abscisses, donne le point 4. Et ainsi de suite jusqu'au critère d'arrêt qui consiste à ce que l'algorithme n'évolue plus beaucoup, c'est-à-dire $|x_n - x_{n-1}| \leq \epsilon$ pour un niveau de précision ϵ choisi.

Autres méthodes Les autres méthodes sont des légères modifications des quatre méthodes précédentes ou des combinaisons de plusieurs de ces méthodes.
La méthode de la fausse position s'apparente à la méthode de dichotomie, à la seule différence que l'intervalle n'est pas coupé en deux mais en un point donné par la méthode de la sécante.
La méthode de Müller est une méthode de la sécante où la fonction f, inconnue, est approchée par interpolation quadratique.
Comme son nom l'indique, la méthode d'interpolation quadratique inverse fournit une interpo-

lation de la bijection réciproque de f.
La méthode de Brent est une combinaison de la méthode de dichotomie, de la méthode de la sécante et de l'interpolation quadratique inverse. A chaque itération, la méthode susceptible d'approcher au mieux le zéro est choisie.

2.1.3 Méthodes de résolution de systèmes d'équations

Dans le cas des systèmes d'équations, on dispose de m équations et n inconnues. Dans ce cas, on considère n entrées $x_1, \ldots, x_n$ et m fonctions $f_1, \ldots, f_m$ telles que

$$\begin{cases} f_1(x_1, \ldots, x_n) = b_1 \\ \vdots \\ f_m(x_1, \ldots, x_n) = b_m. \end{cases} \tag{IV.7}$$

ce qui équivaut à résoudre $\phi(x) = b$, où $x = (x_1, \ldots, x_n)$, $b = (b_1, \ldots, b_m)$ et $\phi = (f_1, \ldots, f_m)$. La solution d'un tel système doit être unique. Pour les systèmes d'équations linéaires, chaque fonction du système (IV.7) s'écrit

$$f_j(x) = \sum_{i=1}^{n} \lambda_i^j x_i.$$

Le système peut se réécrire $Ax = b$, où A est la matrice $(m \times n)$ qui contient les coefficients $\lambda_i^j, 1 \leq i \leq n, 1 \leq j \leq m$, x le vecteur de taille n contenant les x_i et b le vecteur de taille m contenant les b_j. Les systèmes linéaires sur-déterminés $(m > n)$ sont en général résolus par moindres carrés

$$min_x ||Ax - b||^2. \tag{IV.8}$$

La résolution des moindres carrés linéaires a été largement étudiée dans (Lawson and Hanson, 1974) et (Björck, 1990) qui décrivent notamment les différentes décompositions et factorisations de la matrice $A(m \times n)$ telle que $Ax = b$. Il s'agit de la décomposition en valeurs singulières (Golub and Loan, 2012) et la factorisation QR.

Décomposition en valeurs singulières Cette décomposition consiste à exprimer la matrice $A(m \times n)$ comme $A = U\Sigma V$, où $U(m \times m)$ et $V(n \times n)$ sont des matrices orthogonales, c'est-à-dire que $UU^t = U^tU = I_m$ et $VV^t = V^tV = I_n$, et $\Sigma(m \times n)$ est une matrice bloc qui

s'écrit de la façon suivante

$$\left(\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right),$$

où $\sigma_1 \geq \ldots \geq \sigma_r$, avec $r = rang(A)$.
Si l'on note $U = (u_1, \ldots, u_m), V = (v_1, \ldots, v_n)$ les colonnes des matrices U et V, alors les vecteurs u_j et v_j sont, respectivement, les vecteurs singuliers droits et gauches associés à la valeur singulière σ_j.
Comme Σ est diagonale, alors la résolution du problème de moindres carrés se fait composante par composante dans les bases $(u_1, \ldots, u_m)$ et $(v_1, \ldots, v_n)$.

Factorisation QR Cette factorisation utilise un type bien particulier de matrices : les matrices de Householder qui sont symétriques et orthogonales. La factorisation QR consiste à exprimer la matrice $A(m \times n)$ de rang n comme $A = QR$, où Q est une matrice orthogonale et R une matrice triangulaire supérieure. Outre avec la méthode de Householder, cette décomposition peut être obtenue par la méthode de Givens ou celle de Gram-Schmidt.
Le système $Ax = b$ devient alors $QRx = b$. La résolution repose sur la définition du vecteur auxiliaire $y = Rx$ puisqu'il s'agit de résoudre $Qy = b$ dans un premier temps puis $Rx = y$. Comme Q est orthogonale, alors $y = Q^t b$. Comme R est triangulaire, la résolution de la seconde équation est simple.

Les cas sous-déterminés ($m < n$) sont mal posés et doivent être régularisés, souvent en ajoutant $n - m$ fonctions qui peuvent être des contraintes sur x. Le problème devient alors la résolution d'un système à n équations et n inconnues. L'existence et l'unicité de la solution à un tel système est assuré par la règle de Cramer. On note $\phi : \mathbb{R}^n \to \mathbb{R}^n$ la fonction correspondant au système (IV.7), c'est-à-dire telle que $\phi(x) = 0$.

Théorème 2 (Règle de Cramer)**.** *Une condition nécessaire et suffisante pour que le système linaire IV.7 à n équations et n inconnues admette une solution unique pour tout $b \in \mathbb{R}^n$ est de manière équivalente*

- *$det(A) \neq 0$,*
- *$Ker(\phi) = \{0\}$,*
- *$rang(\phi) = dim(Im(\phi)) = n$,*
- *les vecteurs lignes et vecteurs colonnes de A sont indépendants,*

- *la seule solution de* $Ax = 0$ *est* $x = 0$.

Il existe de nombreuses méthodes de résolution des systèmes linéaires, comme :
- les méthodes directes du pivot de Gauss ou des factorisations de Crout et de Cholesky,
- la méthode de Gauss-Seidel,
- les méthodes de relaxation,
- la méthode Jacobi,
- la descente de gradient,
- le gradient conjugué.

Méthodes directes Les méthodes directes permettent d'obtenir une solution exacte en un nombre fini d'opérations. Le pivot de Gauss permet de ramener la résolution d'un système linéaire quelconque à un système triangulaire supérieur par triangularisation de la matrice A. Si A est symétrique définie positive, alors A est forcément triangularisable et la méthode du pivot est équivalente à la factorisation $A = LU$, où L est une matrice triangulaire inférieure avec une diagonale unité et U une matrice triangulaire supérieure. Pour A symétrique, on peut utiliser cette symétrie pour obtenir une factorisation de Crout, $A = LDL^T$, avec D diagonale. Si A est symétrique définie positive, alors on peut utiliser la factorisation de Cholesky, $A = LL^T$, où L n'est plus à diagonale unité. Par ces méthodes, on obtient un système triangulaire supérieur équivalent au système initial et qui est résolu explicitement par un processus de remontée. Le lecteur intéressé par les méthodes directes pourra se référer à (Lascaux and Théodor, 1987) et (Lucquin et al., 1998).

Méthode de Gauss-Seidel La méthode de Gauss-Seidel consiste à décomposer $A = D - E - F$, avec D la matrice diagonale constituée des éléments diagonaux de A et E (resp. F) est une matrice triangulaire inférieure stricte (resp. supérieure stricte) composée des éléments strictement sous-diagonaux (resp. sur-diagonaux) de A. La méthode est itérative

$$\begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = (D-E)^{-1}Fx^{(k)} + (D-E)^{-1}b. \end{cases} \tag{IV.9}$$

Dans (Jeffreys and Jeffreys, 1999), davantage d'informations sont fournies sur la méthode.

Les méthodes de relaxation sont une généralisation de la méthode de Gauss-Seidel. On considère

un paramètre w et la méthode itérative

$$\begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = (D - wE)^{-1}(wF + (1-w)D)x^{(k)} + w(D - wE)^{-1}b. \end{cases} \tag{IV.10}$$

Le cas $w = 1$ correspond à la méthode de Gauss-Seidel.

Méthode de Jacobi La méthode de Jacobi traite les équations du système de manière indépendante. Le terme général de la méthode itérative est

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b.$$

Cette méthode est en général moins efficace que celle de Gauss-Seidel. Elle a été étudiée, entre autres, dans (Varga, 1962).

Méthodes de descente Les méthodes de descente sont des méthodes itératives qui utilisent l'équivalence entre la résolution de $Ax = b$, avec A symétrique définie positive et la minimisation de $J(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$, où $\langle ., .\rangle$ désigne le produit scalaire. Ces méthodes sont basées sur le calcul de la solution comme limite d'une suite minimisante de la forme quadratique J. Cette suite est construite comme une suite récurrente :

$$\begin{cases} x^{(0)} \text{ donné} \\ x^{(k+1)} = x^{(k)} - \alpha_k d^{(k)}, \end{cases} \tag{IV.11}$$

avec $d^{(k)} \in \mathbb{R}^n$ le vecteur indiquant la direction de descente à l'étape k et $\alpha_k \in \mathbb{R}$ le coefficient permettant de minimiser J dans la direction $d^{(k)}$. La méthode de Gauss-Seidel est une méthode de descente correspondant aux choix des axes de coordonnées comme directions successives de descente. Dans le cas de la méthode du gradient, on choisit comme direction de descente la direction du vecteur gradient de J au point $x^{(k)}$.

Méthode du gradient conjugué La méthode du gradient conjugué (chap. 8 dans (Luenberger, 1973)) est utilisée pour les systèmes linéaires dont la matrice est symétrique définie positive. Le principe de la méthode repose sur la recherche de directions successives permettant d'atteindre la solution exacte x_* du système étudié. Comme la matrice A est symétrique définie positive, on peut définir le produit scalaire sur $\mathbb{R}^n$: $\langle u, v\rangle_A = u^T Av$. Deux éléments $u, v \in \mathbb{R}^n$ sont dits A-conjugués si $u^T Av = 0$. La méthode du gradient conjugué consiste à construire

une suite $(p_k)_{k\in\{1,\dots,n\}}$ de n vecteurs A-conjugués. La suite $p_1, p_2, \dots, p_n$ forme une base de $\mathbb{R}^n$. La solution exacte x_* peut se décomposer de la manière suivante : $x_* = \alpha_1 p_1 + \dots + \alpha_n p_n$ où $\alpha_k = \frac{p_k^T b}{p_k^T A p_k}, k = 1, \dots, n$.
La solution exacte x_* peut également être vue comme l'argument minimum de la fonction $J(x) = \frac{1}{2}x^T Ax - b^T x, x \in \mathbb{R}^n$. On a donc clairement $\nabla J(x) = Ax - b, x \in \mathbb{R}^n$, d'où $\nabla J(x_*) = 0_{\mathbb{R}^n}$. On définit le résidu du système d'équation $r_k = b - Ax_k = -\nabla J(x_k)$ qui représente la direction du gradient de la fonction J en x_k. La nouvelle direction de descente p_{k+1} suit celle du résidu, modulo sa A-conjugaison avec p_k. On a alors $p_{k+1} = r_k - \frac{p_k^T A r_k}{p_k^T A p_k} p_k$.

Méthodes de gradient en non-linéaire Les méthodes de gradient se généralisent au cas non-linéaire où il s'agit de minimiser J strictement convexe avec le gradient ∇J non-linéaire. La détermination du pas optimal α_k à l'itération k se fait alors par une méthode de recherche de l'argument α_k qui minimise la fonction $J(x^{(k)} + \alpha \nabla J(x^{(k)}))$. On est ramené à un problème en dimension 1 pour lequel diverses techniques existent, en particulier les algorithmes de recherche linéaire suivant différentes règles (Amijo, Wolfe, Goldstein-Price). L'extension de la méthode du gradient conjugué au cas non-linéaire nécessite aussi la définition des directions de descente conjuguées. L'algorithme le plus efficace pour cela est celui de Polak-Ribière.

Dans (Björck, 1990) sont également proposées des méthodes de résolution des moindres carrés non-linéaires, notamment les méthodes de type Newton. Pour résoudre les moindres carrés non-linéaires, on trouve aussi la méthode de Levenberg-Marquardt.

Méthodes de Newton Nous avons vu que la méthode de Newton-Raphson pouvait se généraliser à la résolution de systèmes d'équations non-linéaires pour $m = n$. La méthode consiste à considérer l'itération principale comme

$$x_{k+1} = x_k + d,$$

où d est la solution de l'équation suivante

$$J(x_k).d = -f(x_k),$$

avec J la matrice jacobienne de f

$$J(x) = \left(\frac{\partial f_i}{\partial x_j}(x)\right)_{1\le i,j\le n}.$$

L'algorithme s'arrête dès que $d(x_k, x_{k-1}) \leq \epsilon$ ou $d(.,.)$ est une distance définie sur $\mathbb{R}^n$ et ϵ la précision choisie par l'utilisateur. Le principal problème de cette méthode est le calcul et l'inversion de la jacobienne à chaque itération de l'algorithme. C'est pourquoi il est parfois préférable d'utiliser une méthode de quasi-Newton où la jacobienne J est approchée par une matrice B mise à jour à chaque itération. Il existe plusieurs méthodes de quasi-Newton comme la méthode de Broyden (Broyden, 1965). Dans ce cas la mise à jour de la matrice jacobienne se fait avec des matrices de rang 1. Étant donnée une approximation $B^{(k)}$ de la matrice jacobienne à l'itération k on calcule l'approximation de l'itération $k+1$ comme

$$B^{(k+1)} = B^{(k)} + \frac{\left((f(x_{k+1}) - f(x_k)) - B^{(k)} d_{(k)}\right) d'_{(k)}}{d'_{(k)} d_{(k)}},$$

où $d_{(k)}$ vérifie $B^{(k)} d_{(k)} = -f(x_k)$. Si on note x_0 le point initial de l'algorithme, alors on pose $B^{(0)} = \nabla f(x_0)$.

La méthode de Newton, ainsi que la descente de gradient, font partie d'une classe de méthodes : l'optimisation locale. Notons que dans ce cas, la méthode de Newton résout $g'(x) = 0$ où g' est la dérivée de la fonction à maximiser ou à minimiser, comme les moindres carrés.

2.1.4 Méthodes d'optimisation globale

Pour résoudre un problème de moindres carrés, il existe aussi des méthodes d'optimisation globale comme les algorithmes évolutionnaires, le recuit simulé et la recherche tabou. Ces méthodes font partie d'une famille générale d'algorithmes d'optimisation : les métaheuristiques. Ce sont généralement des algorithmes stochastiques itératifs, qui progressent vers un optimum global par échantillonnage de la fonction objectif. Ils se comportent comme des algorithmes de recherche, tentant d'apprendre les caractéristiques d'un problème afin d'en trouver une approximation de la meilleure solution (d'une manière proche des algorithmes d'approximation). Il existe un grand nombre de métaheuristiques différentes, allant de la simple recherche locale à des algorithmes complexes de recherche globale. Ces méthodes sont répertoriées et classées dans un diagramme présenté à la Figure IV.4. Les métaheuristiques peuvent être classées de nombreuses façons. C'est pourquoi certaines méthodes sont présentées à cheval entre plusieurs classes. Une telle méthode peut être placée dans l'une ou l'autre classe, selon le point de vue adopté.
Une première classification consiste à distinguer les notions de parcours et de population. Les métaheuristiques les plus classiques sont celles fondées sur la notion de parcours. Dans ce cas,

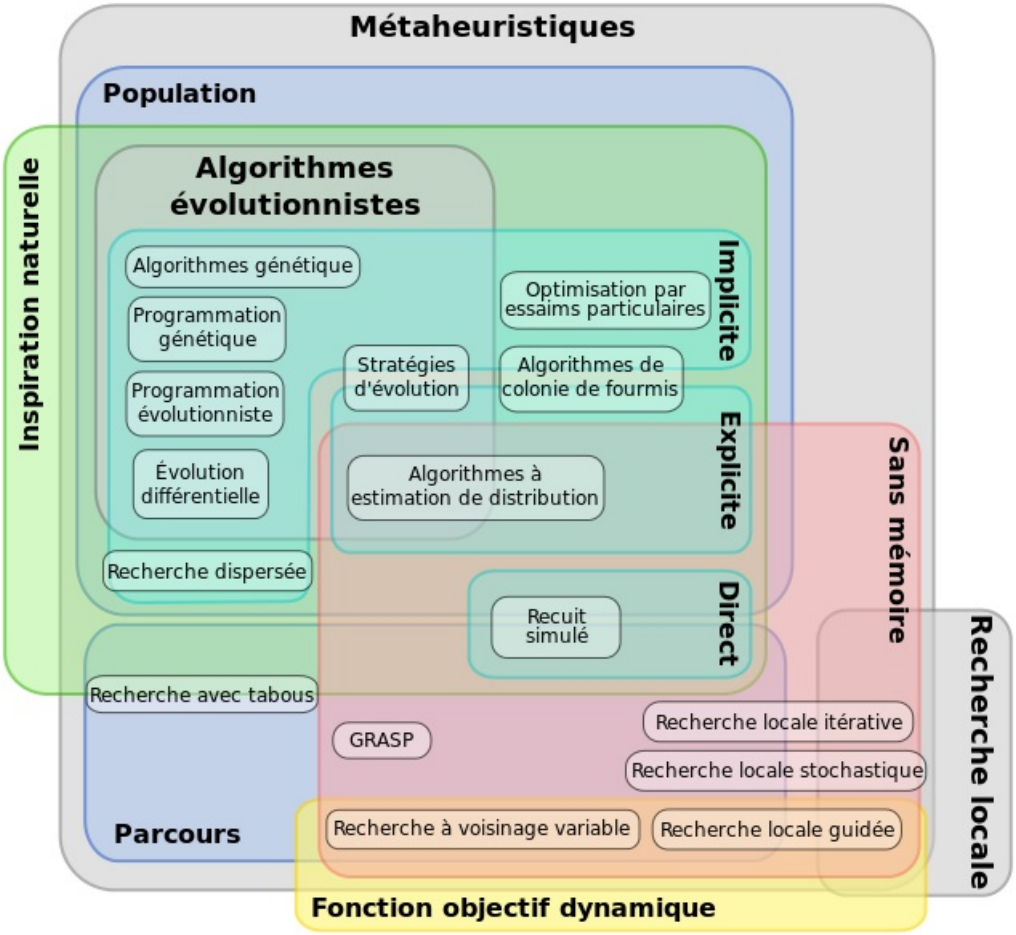

Figure IV.4 – Classifications des métaheuristiques

une seule solution évolue à chaque itération sur l'espace de recherche. La notion de voisinage est alors primordiale. Les plus connues dans cette classe sont le recuit simulé, la recherche avec tabous, la recherche à voisinage variable, la méthode GRASP (*Greedy Randomized Adaptive Search Procedure*) ou encore les méthodes de bruitage. L'autre approche utilise la notion de population. Dans ce cas, un ensemble de solutions sont considérées en parallèle, à chaque itération. On y trouve les algorithmes génétiques, l'optimisation par essaims particulaires, les algorithmes de colonies de fourmis. Ces deux dernières méthodes sont fondées sur le principe qu'un groupe d'individus peu intelligents peut posséder une organisation globale complexe.
Une autre classification porte sur le fait que l'algorithme a une mémoire ou pas. En effet, les métaheuristiques utilisent l'historique de leur recherche pour guider l'optimisation aux itérations suivantes. Dans le cas le plus simple, elles considèrent l'état de la recherche à une itération

donnée pour déterminer la prochaine itération. Il s'agit alors d'un processus de décision markovien et on parlera de méthode sans mémoire. C'est le cas de la plupart des méthodes de recherche locale. Beaucoup de métaheuristiques utilisent une mémoire plus évoluée comme la recherche tabous.
Les métaheuristiques sont des méthodes itératives utilisant un échantillonnage de la fonction objectif comme base d'apprentissage. Le choix de cet échantillonnage se fait en général via une distribution de probabilités. L'utilisation de cette distribution donne lieu à trois classes de métaheuristiques :

- les méthodes implicites : la distribution de probabilité n'est pas connue a priori,
- les méthodes explicites : une distribution de probabilité choisie à chaque itération,
- les méthodes directes : ni implicites ni explicites.

Parmi les méthodes implicites, on trouve les algorithmes génétiques, où le choix de l'échantillonnage entre deux itérations ne suit pas une loi donnée, mais est fonction de règles locales. Dans la classe des méthodes explicites, on trouve les algorithmes à estimation de distribution. Dans cette classification, le recuit simulé représente la classe des méthodes directes. Dans cette méthode, on peut considérer qu'il échantillonne la fonction objectif en l'utilisant directement comme distribution de probabilité (les meilleures solutions ayant une probabilité plus grande d'être tirées).
Une dernière classification repose sur une propriété de la fonction objectif. On peut séparer les métaheuristiques selon qu'elles utilisent une fonction objectif statique (qui demeure inchangée tout au long de l'optimisation) ou dynamique (quand la fonction objectif est modifiée au cours de la recherche). Dans la seconde classe, on trouve notamment la recherche à voisinage variable et la recherche locale guidée.

Pour plus d'informations sur les métaheuristiques, le lecteur intéressé pourra se référer à (Dréo et al., 2003), (Blum and Roli, 2003), (Talbi, 2009) ou encore (Glover and Kochenberger, 2003). Nous présentons ici les algorithme évolutionnaires, le recuit simulé et la recherche avec tabous.

Algorithmes évolutionnaires Les algorithmes évolutionnaires sont des méthodes suggérées par le paradigme Darwinien de l'évolution. Le principe de variation et de sélection peut être considéré comme le principe fondamental de l'évolution Darwinienne. Ce principe, associé au changement de chaque génération (la mutation), fait naître les composants fondamentaux de la boucle de l'algorithme évolutionnaire. Le principe général de fonctionnement de ces algorithmes est représenté par le schéma de la Figure IV.5. Ces algorithmes commencent par une étape d'initialisation avec l'évaluation d'un certain nombre de points appelés individus. Ils

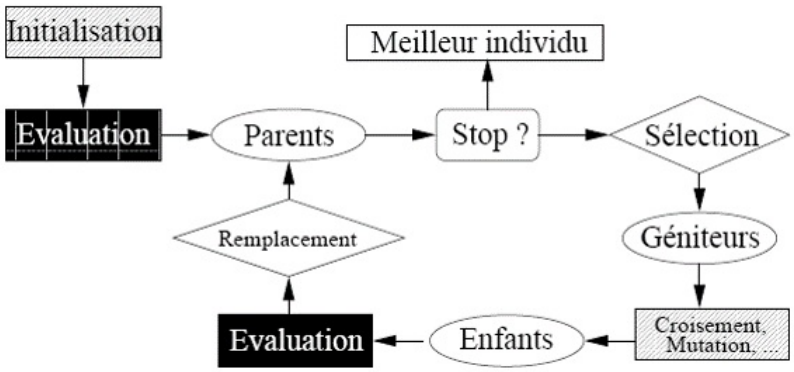

Figure IV.5 – Schéma de fonctionnement général d'un algorithme évolutionnaire

constituent une population. Si le critère d'arrêt n'est pas satisfait, on sélectionne les meilleurs individus qui vont devenir des géniteurs. Ces parents vont générer un certain nombre d'enfants grâce à des opérateurs stochastiques de variation. Ces enfants sont évalués pour constituer avec les anciens parents une nouvelle population, sur laquelle on va de nouveau effectuer les différentes étapes. L'évolution stoppe quand le niveau de performance souhaité est atteint ou qu'un nombre fixé de générations s'est écoulé sans améliorer l'individu le plus performant.
On distingue plusieurs types d'algorithmes évolutionnaires :

- les algorithmes génétiques inspirés des mécanismes de l'évolution naturelle,
- la programmation génétique, extension des algorithmes génétiques dans laquelle les individus sont des programmes,
- les systèmes de classifieurs, mécanismes d'apprentissage basés sur un ensemble de règles condition/action. Chaque règle est notée en fonction du résultat de l'action produite et un algorithme génétique est utilisé pour générer de nouvelles règles,
- les stratégies d'évolution (*Evolution Strategy* en anglais, ES), algorithmes itératifs dans lesquels un parent génère un enfant (1+1)-ES. Le meilleur des deux survit et devient le parent de la génération suivante. La génération de ce processus a donné les algorithmes $(\mu + \lambda)$-ES dans lesquels μ parents génèrent λ enfants. Les μ meilleurs survivent.

Des précisions sur ces méthodes sont fournies dans (Bäck, 1996) et (Goldberg et al., 1994).

Recuit simulé Le recuit simulé est un algorithme d'optimisation inspiré d'un processus physique d'amélioration de la qualité d'un solide en métallurgie. Le principe général de la méthode est d'effectuer un mouvement selon une loi de probabilité dépendant de la qualité des voisins (les meilleurs ont les probabilités les plus élevées). En plus de cela, un paramètre de température, T, est considéré pour faire le lien entre le mouvement et la fonction de coût à

minimiser. Ce paramètre est élevé au début de l'algorithme (les voisins ont à peu près la même probabilité d'être acceptés), puis diminue (un mouvement qui dégrade la fonction de coût a une faible probabilité d'être choisi), jusqu'à tendre vers 0 (aucune dégradation de la fonction de coût n'est acceptée).
Pour un point donné de l'algorithme, on génère un voisin dont l'écart au point en cours, en terme de performance, est évalué. Un critère de Métropolis permet de déterminer si le voisin est conservé selon l'amélioration qu'il apporte à l'algorithme. Ce critère dépend de l'écart et de la température à ce niveau de l'algorithme.
Le critère d'arrêt repose sur le manque d'évolution de l'algorithme et sur le fait que le pourcentage de voisins acceptés devienne faible.
Une description plus détaillée des particularités de la méthode est fournie dans (Kirkpatrick et al., 1983) et (Siarry and Dreyfus, 1988).

Recherche avec tabous Cette méthode est une métaheuristique itérative qui consiste à explorer le voisinage d'une position donnée et à choisir la position dans ce voisinage qui minimise la fonction objectif. La force de la recherche avec tabous est que cette opération peut conduire à augmenter la valeur de la fonction (dans un problème de minimisation) : c'est le cas lorsque tous les points du voisinage ont une valeur plus élevée. C'est à partir de ce mécanisme que l'on sort d'un minimum local et que la méthode peut tendre vers l'optimum global. De plus, cette méthode est une métaheuristique avec mémoire. Ceci évite que l'algorithme ne retombe dans l'optimum local dont il vient de sortir. Cette qualité de mémoire consiste simplement à interdire (d'où le nom de tabou) de revenir sur les dernières positions explorées. Ces positions sont stockées dans une liste tabou dont la taille est choisie par l'utilisateur. De nombreuses variantes de la méthode existent, principalement au niveau de la définition du voisinage et de la manière de gérer la mémoire (court, moyen ou long terme). Les démonstrations de convergence pour la recherche tabou existent, mais supposent des conditions strictes, rarement présentes en pratique. Le lecteur intéressé par cette méthode pourra se référer à (Glover and Kochenberger, 2003).

Les autres méta-heuristiques ont également fait l'objet de recherches. Les algorithmes de colonies de fourmis ont été étudiés dans (Bonabeau et al., 1999), la méthode GRASP dans (Resende, 2009) et (Angel-Bello et al., 2006), l'optimisation par essaims particulaires dans (Eberhart et al., 2001), les algorithmes à estimation de distribution dans (Larranaga and Lozano, 2002), la recherche à voisinage variable dans (Brimberg et al., 2010) et (Altınel et al., 2011), la recherche dispersée dans (Glover, 1977), la recherche locale itérative dans (Lourenço et al., 2010), la re-

cherche locale stochastique dans (Stutzle and Hoos, 2005) et la recherche locale guidée dans (Talbi, 2009).

La recherche est très active dans ce domaine, il est donc difficile d'établir une liste exhaustive des différentes métaheuristiques d'optimisation. La littérature spécialisée montre un grand nombre de variantes et d'hybridations entre méthodes, particulièrement dans le cas des algorithmes évolutionnaires. Cependant, nous pouvons citer d'autres métaheuristiques, plus ou moins connues, qui ne sont pas répertoriées dans le diagramme de la Figure IV.4 :

- l'algorithme du kangourou,
- la méthode de Fletcher et Powell,
- la méthode du bruitage,
- la tunnelisation stochastique,
- l'escalade de collines à recommencements aléatoires,
- la méthode de l'entropie croisée,
- l'algorithme de recherche d'harmonie

Nous présentons la méthode de l'entropie croisée qui a la particularité de résoudre des problèmes à plusieurs optima globaux.

Méthode de l'entropie croisée Nous nous sommes particulièrement intéressés à cette méthode récente permettant de réaliser une optimisation globale : la méthode de l'entropie croisée (*cross-entropy* en anglais). Attribuée à Rubinstein (Rubinstein and Kroese, 2013), la méthode de l'entropie croisée est une méthode générale d'optimisation de type Monte-Carlo (combinatoire ou continue) et d'échantillonnage préférentiel. Conçue à l'origine pour la simulation d'évènements rares, elle permet de résoudre des problèmes d'optimisation combinatoire par simple modification de l'entropie croisée. Elle repose notamment sur la traduction du problème d'optimisation déterministe en un problème stochastique associé et permet ainsi l'utilisation des techniques de simulation d'évènements rares.
La procédure itérative de la méthode de l'entropie croisée consiste à répéter, à chaque itération, deux actions :

1. générer aléatoirement un échantillon de données (trajectoires, vecteurs, etc.) selon un mécanisme spécifique,
2. mettre à jour les paramètres du mécanisme de génération aléatoire à partir de l'échantillon de données pour produire un meilleur échantillon à l'itération suivante (étape impliquant la minimisation de l'entropie croisée ou la divergence KL).

L'entropie croisée et la divergence de Kullback-Leibler sont liées, comme nous pouvons le voir

dans la Définition 6.

Définition 6. *Soient 2 distributions p et q sur le même espace probabilisé $\mathcal{X}$. Leur entropie croisée est définie par*

$$H(p,q) = \mathbb{E}_p[-\log q] = H(p) + D_{KL}(p||q), \tag{IV.12}$$

où $H(p)$ est l'entropie de p et $D_{KL}(p||q)$ la divergence de Kullback-Leibler entre p et q.
Dans un cas discret,

$$\begin{aligned} H(p) &= -\sum_x p(x)\log p(x), \\ D_{KL}(p||q) &= \sum_x p(x)\log\frac{p(x)}{q(x)}, \\ H(p,q) &= -\sum_x p(x)\log q(x). \end{aligned}$$

Dans un cas continu,

$$\begin{aligned} H(p) &= -\int_X p(x)\log p(x)dx, \\ D_{KL}(p||q) &= \int_X p(x)\log\frac{p(x)}{q(x)}, \\ H(p,q) &= -\int_X p(x)\log q(x)dx. \end{aligned}$$

Le but de la méthode est de maximiser une fonction objectif $S(x)$ sur tous les éléments $x \in \mathcal{X}$. On note μ^* le maximum

$$\mu^* = \max_{x\in\mathcal{X}} S(x). \tag{IV.13}$$

La randomisation de ce problème déterministe consiste à définir une famille paramétrique de densités de probabilité $\{f(.,v), v \in V\}$ sur $\mathcal{X}$. Soit X un vecteur aléatoire de densité $f(.,v_0), v_0 \in V$ et γ un paramètre connu ou inconnu. On pose

$$l(\gamma) := \mathbb{P}_{v_0}(S(X) \geq \gamma) = \mathbb{E}_{v_0}[I_{\{S(X)\geq\gamma\}}] \tag{IV.14}$$

la quantité à estimer pour un niveau γ donné. L'estimation de cette quantité s'appelle le problème stochastique associé. La résolution du problème (IV.13) est équivalente à celle du problème (IV.15).

$$\min_{v\in V} H(I_{\{S(X)\geq\gamma\}}, f(.,v)), \tag{IV.15}$$

pour un niveau γ donné avec H défini en IV.12. L'idée de la méthode est de générer une suite de densités paramétriques $f(.,v_0), f(.,v_1), f(.,v_2), \ldots$. Cette suite tend vers la densité optimale théorique qui est la densité dégénérée au point optimal, $f(.,v^*)$. La loi est souvent considérée comme étant une loi normale. Les paramètres de la loi constituent donc un vecteur à deux éléments, la moyenne et l'écart-type. Une suite de paramètres est alors construite $(v_t) := \{(\mu_t, \sigma_t)\}, t \geq 1$ et doit tendre vers le vecteur optimal $(\mu^*, \sigma^*) = (\mu^*, 0)$, les paramètres de la densité optimale.
Les principales étapes de la méthode sont décrites dans l'Algorithme 1.

Initialisation
$\rho \leftarrow 10^{-2}$.
$v_0 \leftarrow \mu$.
$t \leftarrow 0$.
Tant que (le critère d'arrêt n'est pas satisfait) **faire**
  $t \leftarrow t+1$.
  Soit $X_1, \ldots, X_N$ un échantillon aléatoire tiré à partir de $f(.,v_{t-1})$.
  Déterminer μ_t le quantile empirique d'ordre $1-\rho$ de $S(X)$ sous v_{t-1}.
  Déterminer v_t tel que $v_t = \arg\max_v \frac{1}{N}\sum_{i=1}^{N} I_{\{S(X_i)\geq\mu_t\}} \ln f(X_i, v)$.
Fait

Algorithme 1 – Algorithme général de la méthode de l'entropie croisée

Dans un cas non-contraint, nous avons testé la méthode sur le problème de maximisation de la fonction

$$S(x) = e^{-(x-2)^2} + 0.8e^{-(x+2)^2}, x \in \mathbb{R}.$$

Cette fonction possède un maximum local et un maximum global, comme le montre la Figure IV.6. A partir d'un vecteur de paramètres initial $(\mu_0, \sigma_0) = (-6, 100)$, on obtient rapidement l'optimum $(\mu^*, \sigma^*) = (2, 0)$. Les différentes étapes de l'algorithme sont représentées à la Figure IV.7. La méthode permet également de résoudre des problèmes à plusieurs extrema. Par exemple, nous tentons de résoudre le problème de maximisation de la fonction

$$S(x) = 150e^{-\frac{x^2}{5}} + 2x^2 - \frac{x^4}{200},$$

représentée à la Figure IV.8. Dans ce cas, on ne considère plus une loi de probabilité mais un mélange de deux lois. Chacune doit tendre vers un des deux maxima. En quelques itérations, l'algorithme converge vers les deux solutions, comme le montre la Figure IV.9. La méthode permet de résoudre des problèmes moins réguliers comme l'optimisation des fonctions trigo-

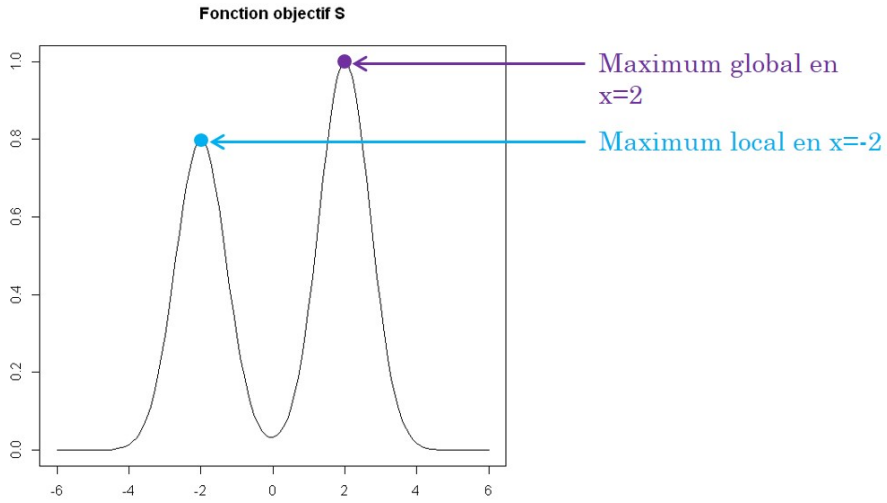

Figure IV.6 – Résolution d'un cas non-contraint avec la méthode d'entropie croisée

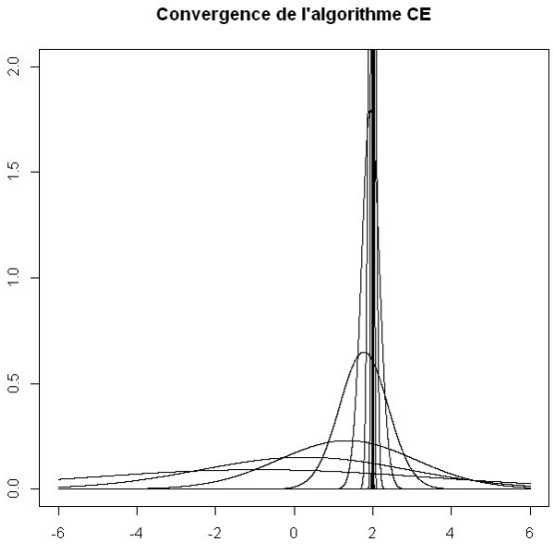

Figure IV.7 – Évolution de l'algorithme d'entropie croisée sur le cas non-contraint

nométriques, de Rosenbrock ou de Hougen. Dans les cas contraints, la prise en compte des contraintes se fait soit par acceptation-rejet, soit par pénalité.
Cette méthode est simple et converge rapidement et indépendamment du point de départ. Dans un cas à plusieurs extrema, il faut choisir le nombre de densités, c'est-à-dire connaître a priori le nombre d'optima, ce qui n'est pas forcément le cas. De plus, en pratique, de nombreux paramètres sont à régler, principalement lorsque le problème est contraint. Enfin, la convergence de la méthode n'a été démontrée que sur des cas simples.

Les méthodes classiques sont utilisées dans le cas des fonctions de $\mathbb{R}$ dans $\mathbb{R}$, les méthodes

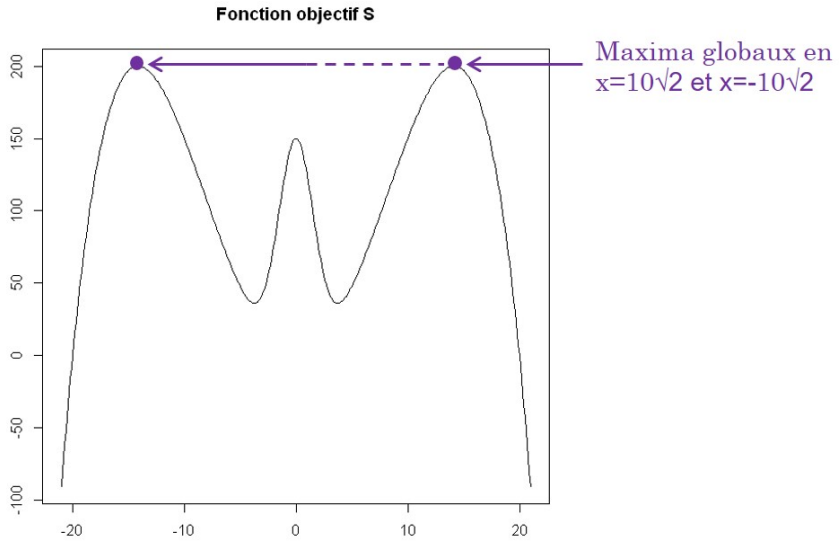

Figure IV.8 – Fonction à deux maxima globaux

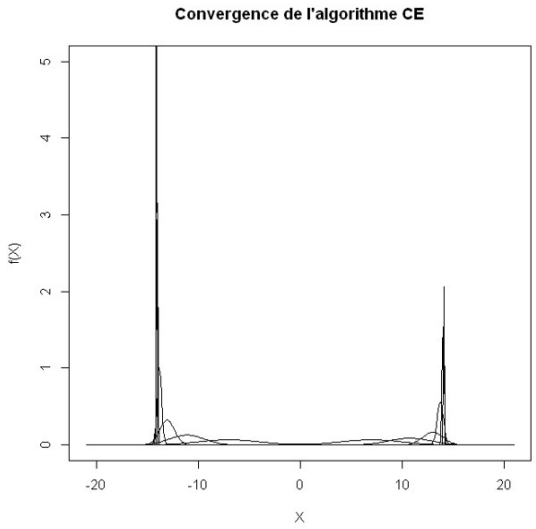

Figure IV.9 – Évolution de l'algorithme d'entropie croisée sur le cas à deux optima

de résolution de systèmes d'équations dans le cas des fonctions de $\mathbb{R}^m$ dans $\mathbb{R}^n$, où $m \geq n$ linéaires ou non. Les méthodes d'optimisation permettent de trouver une solution unique à un problème général pour une fonction continue à valeurs dans $\mathbb{R}$, voire dans $\mathbb{R}^k, k \geq 1$, avec les techniques d'optimisation multi-objectifs. Dans tous les cas, ce sont des problèmes bien posés ou régularisés. Cependant, notre problème consiste à étudier les antécédents d'une fonction de $\mathbb{R}^d$ dans $\mathbb{R}$. L'ensemble de ces antécédents n'est donc pas fini et n'est souvent pas continu. Il s'agit d'un problème inverse mal posé pour lequel nous n'avons pas suffisamment de connaissances pour le régulariser. Il nous intéresse d'ailleurs davantage d'obtenir plusieurs solutions plutôt

qu'une seule sous certaines contraintes a priori.

2.2 Résolution de problèmes inverses mal posés

La résolution d'un tel problème avec les méthodes classiques de recherche de zéros nécessiterait de fixer $d-1$ variables pour trouver un d-uplet solution et de réitérer l'algorithme en changeant les $d-1$ variables fixées à des valeurs différentes. Le résoudre avec les méthodes de résolution de systèmes d'équations consisterait à ajouter $d-1$ contraintes afin de rendre le problème bien posé. Ceci devient très contraignant et difficile lorsque d est grand et que la connaissance sur le processus est limité. Nous pourrions utiliser des méthodes d'optimisation qui ne nous conduirait que vers une des solutions alors que nous voulons en obtenir un grand nombre.
Suivant la valeur de la dimension d du domaine d'entrée D, l'ensemble des solutions $S = \{x \in D \subset \mathbb{R}^d : f(x) = 0\}$ sera composé :

- de singletons si $d = 1$,
- de courbes si $d = 2$,
- de surfaces si $d = 3$,
- d'hypersurfaces de dimension $d-1$ pour $d > 3$.

Pour certaines classes d'équations du type $f(x) = a$ où $x \in \mathbb{R}^d, d > 1$, des algorithmes ont été trouvés pour les résoudre. Certains d'entre eux ont été implémentés dans des systèmes de calcul formel. Mais en général, il n'y a pas d'algorithme permettant de résoudre systématiquement de telles équations.
Plusieurs méthodes existent cependant pour résoudre de tels problèmes : l'optimisation locale multi-start, une méthode déterministe par recherche de grilles et deux méthodes stochastiques, l'une basée sur une recherche probabiliste via un modèle statistique, l'autre sur du MCMC (chaînes de Markov par Monte-Carlo). A notre connaissance, seules ces méthodes permettent de résoudre de tels problèmes.

2.2.1 Optimisation locale multi-start

L'optimisation locale multi-start (György and Kocsis, 2011) consiste dans un premier temps à choisir une méthode d'optimisation locale comme une méthode de Newton, de descente ou de gradient. On choisit également un plan qui quadrille bien le domaine d'entrée. Chaque point de ce plan représente un point de départ pour une optimisation locale. Le choix du plan est très important et conditionne le nombre de solutions obtenues par la méthode. On choisit le plus souvent un plan latin hypercube qui satisfait un critère de bonne répartition appelé maximin.

Certains points de départs peuvent conduire à une même solution, d'autres ne permettent pas d'en trouver une satisfaisante. De plus, c'est une méthode coûteuse d'abord à cause du plan initial, ensuite par rapport à la méthode d'optimisation. Enfin, la qualité et le nombre de points solutions dépend du plan et peuvent différer d'un plan à l'autre.

2.2.2 Recherche déterministe de grilles

Cette méthode est qualifiée de déterministe car aucun élément aléatoire n'est utilisé à chaque étape de la méthode. En effet, le choix de nouveaux points se fait selon des règles ou des structures non statistiques. De plus, ni la fonction f ni l'ensemble des solutions S n'est approché par un modèle statistique ou un processus aléatoire. Tous les algorithmes basés sur une recherche déterministe de grilles donnent les mêmes étapes et les mêmes résultats. Seuls les temps de calculs peuvent varier suivant la stratégie choisie.
L'Algorithme 2 décrit de manière générale les principales étapes d'une telle méthode.

Initialisation
Commencer avec un ensemble de points initiaux $E_0 \subset D$ et évaluer f en ces points.
Répéter
A l'itération i, estimer S à partir des points de E_i en lesquels f a été évaluée.
Choisir selon une méthode un ensemble de points $\{x_1^{(i+1)}, \ldots, x_k^{(i+1)}\}$ considérés comme proches de S et évaluer f en ces points.
Poser $E_{i+1} = E_i \cup \{x_1^{(i+1)}, \ldots, x_k^{(i+1)}\}$.
jusqu'à ce que (le critère d'arrêt est satisfait)

Algorithme 2 – Algorithme général de recherche par grilles

Plusieurs algorithmes de recherche déterministe basés sur les grilles sont proposés dans (Miller, 2005), où il est question de recherche d'ensembles de niveaux. On y trouve deux algorithmes de recherche basés sur des cubes et trois basées sur des bords. Dans les deux cas, on commence avec une grille régulière sur D. A l'itération i, on dispose d'un ensemble $C_i \subset E_i$ de points considérés comme proches de S et fournissant une approximation de S. A chaque itération, le nombre de points dans C_i augmente, on note $C_i^{(aug)}$ l'union de C_i et des points ajoutés à l'itération i.
Dans les algorithmes basés sur des cubes, $C_i^{(aug)}$ est un sous-ensemble d'une grille rectangulaire pour laquelle les points voisins sont séparés de w_{i+1}. L'itération $i+1$ de l'algorithme consiste en 5 étapes :

1. *détermination des cubes* : trouver tous les cubes de dimension d et de largeur w_{i+1} formés par les points de $C_i^{(aug)}$. Ces cubes sont notés $\{\mathcal{C}_{j,i}\}_{j=1}^{N(i)}$,
2. *évaluation* : évaluer les points de $C_i^{(aug)} \backslash C_i$,
3. *élagage* : pour chaque cube $\mathcal{C}_{j,i}$, déterminer si f prend des valeurs négatives ou positives en chaque sommet du cube. On ne retient que les cubes pour lesquels f est positive en au moins un sommet et négative en moins un sommet. Une variante consiste à retenir les cubes pour lesquels f est positive en au moins p sommets et négative en au moins p sommets. Ceci constitue une sélection plus restrictive que la première.
   Un point de $C_i^{(aug)}$ est retenu s'il est le sommet d'un cube satisfaisant la condition précédente (ou sa variante). Les autres points sont écartés. L'ensemble des points retenus constitue C_{i+1},
4. *augmentation* : deux versions sont possibles
   (a) *large* : pour chaque cube $\mathcal{C}$ ajouté à l'étape précédente, on ajoute $5^d - 2^d$ points. Il s'agit de 4^d cubes de largeur $w_{i+1}/2$ dont un des sommets est un sommet de $\mathcal{C}$ et situés dans et autour de $\mathcal{C}$. Les sommets de ces cubes permettent de constituer $C_{i+1}^{(aug)}$ avec C_{i+1},
   (b) *faible* : pour chaque cube $\mathcal{C}$ ajouté à l'étape précédente, on ajoute $3^d - 2^d$ points. Il s'agit de 2^d cubes de largeur $w_{i+1}/2$ situés à l'intérieur de $\mathcal{C}$. Les sommets de ces cubes permettent de constituer $C_{i+1}^{(aug)}$ avec C_{i+1}.
   Les deux versions sont représentée sur la Figure IV.10. En trait plein, il s'agit du cube $\mathcal{C}$. En-dehors des sommets du cube, les autres points sont ceux ajoutés suivant la version choisie : (a) à gauche et (b) à droite,
5. *amélioration* : poser $w_{i+2} = w_{i+1}/2$.

Dans les algorithmes basés sur des bords, on dispose comme précédemment des ensemble C_i et $C_i^{(aug)}$ à l'itération i. L'itération $i+1$ de l'algorithme consiste en 5 étapes :

1. *détermination des bords* : trouver tous les bords de largeur w_{i+1} formés par les points de $C_i^{(aug)}$. Il s'agit de trouver toutes les paires de points dans $C_i^{(aug)}$ qui sont voisins dans le quadrillage actuel. On note $B_{j,i}$ le $j$ème bord trouvé à cette étape,
2. *évaluation* : évaluer f en tout point $p \in C_i^{(aug)}$,
3. *élagage* : pour chaque bord $B_{j,i}$, considérer les valeurs de f aux extrémités. On ne retient le bord que si f est positive en une extrémité et négative en l'autre. Un point de $C_i^{(aug)}$ est retenu s'il est l'extrémité d'un bord satisfaisant la condition précédente. Les autres points sont écartés. L'ensemble des points retenus constitue C_{i+1},
4. *augmentation* : trois versions sont possibles
   (a) pour chaque bord $B_{j,i}$ ajouté à l'étape précédente, on ajoute des points sur un hyper-

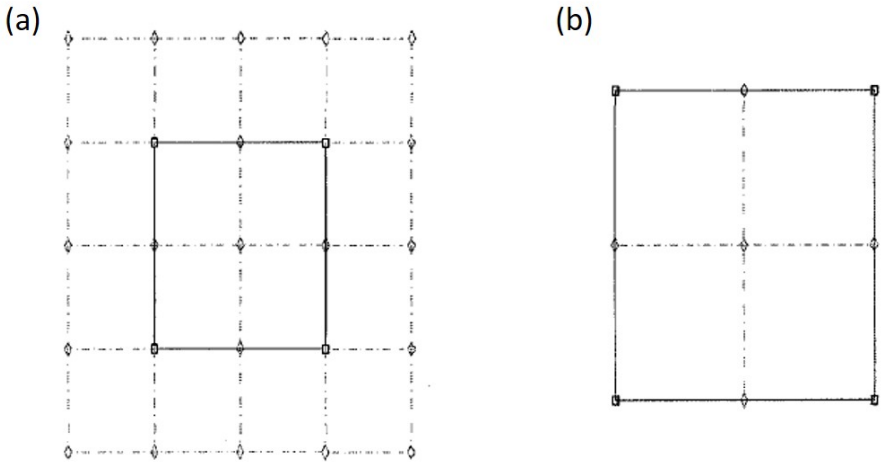

Figure IV.10 – Deux options de l'augmentation du nombre de points dans l'algorithme des cubes

cube de dimension $d-1$ et de largeur $w_{i+1}/2$ dans un plan normal à $B_{j,i}$ et divisant $B_{j,i}$ par 2. Il s'agit de 3^{d-1} points pour chaque bord,

(b) pour chaque bord $B_{j,i}$ ajouté à l'étape précédente, on ajoute des points pour former 2^d cubes, chacun de largeur $w_{i+1}/2$ et avec le milieu de $B_{j,i}$ comme sommet commun. Il s'agit de 3^d-2 points pour chaque bord,

(c) pour chaque bord $B_{j,i}$ ajouté à l'étape précédente, on ajoute des points pour former 2 hyper-cubes de dimension $d-1$ et de largeur $w_{i+1}/3$ dans des hyper-plans parallèles à celui normal à $B_{j,i}$ et divisant $B_{j,i}$ par 3. Il s'agit de $2\times 3^{d-1}$ points pour chaque bord,

avec C_{i+1}, ces points constituent $C_{i+1}^{(aug)}$. Les trois options sont représentées sur la Figure IV.11 avec en trait plein le bord $B_{j,i}$. Les points autres que les extrémités du bord sont ajoutés dans l'algorithme suivant l'option choisie,

5. *amélioration* : poser w_{i+2} selon le choix de la version à l'étape précédente. Pour les deux premiers, $w_{i+2}=w_{i+1}/2$, pour le troisième, $w_{i+2}=w_{i+1}/3$.

Ces méthodes s'appliquent bien dans des cas où S est régulière. Dans (Miller, 2005) des applications sont proposées. Les solutions de ces applications sont un cercle, un cube, une sphère, etc. Pour S irrégulière, la méthode des cubes est plus efficace que celle des bords. Également testée en dimensions supérieures, la recherche déterministe par grilles devient rapidement coûteuse. En effet, la méthode induit le nombre de points évalués à chaque itération. L'utilisateur n'a donc aucun contrôle sur le nombre d'appels à la fonction, qui peut devenir très important pour obtenir une bonne approximation de S. Dans tous les cas, f est supposée concave. De plus, la méthode a pour but d'estimer S. Pour cela, il faut donc un grand nombre de points bien

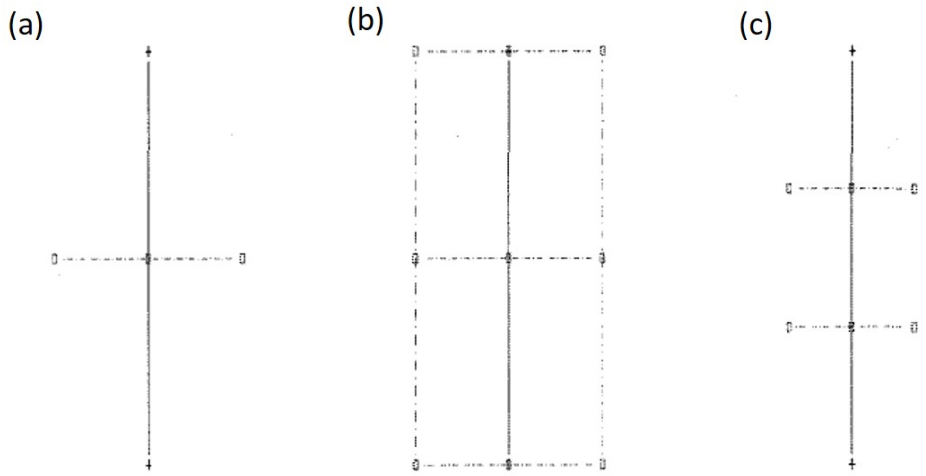

Figure IV.11 – Trois options de l'augmentation du nombre de points dans l'algorithme des bords

répartis au voisinage de S. La proximité des points obtenus à la solution n'est pas maîtrisée non plus et tous les points finaux ne se situent pas tous dans une zone de tolérance choisie. Enfin, la méthode ne peut pas détecter des solutions situées aux bords du domaine étudié et détecte difficilement des solutions isolées.

Face à ces méthodes déterministes, il existe deux méthodes stochastiques permettant de résoudre des problèmes inverses mal posés.

2.2.3 Recherche probabiliste à partir d'un modèle statistique

La méthode de recherche probabiliste est basée sur le même algorithme général, l'Algorithme 2, que la recherche déterministe exposée précédemment. La différence est que les points $\{x_1^{(d+1)}, \ldots, x_m^{(n+1)}\}$ sont choisis en utilisant un modèle statistique. Les données servent à construire une approximation $\hat{f}$ de f, comme par exemple un processus gaussien. La construction d'un tel modèle est décrit en Annexe B. Cette approximation donne une estimation de S : $\hat{S} = \{x \in D : \hat{f}(x) = 0\}$. Les points $\{x_1^{(d+1)}, \ldots, x_m^{(n+1)}\}$ sont choisis parmi des points de $\hat{S}$ selon un critère sur $\hat{f}$. On parle d'un critère d'amélioration prévue (*Expected Improvement* en anglais) expliqué dans (Wagner et al., 2010). L'algorithme ainsi que les critères de sélection des points, de construction du modèle et d'arrêt de l'algorithme sont décrits dans (Miller, 2005). L'algorithme tend à ajouter peu de points à chaque itération, contrairement à la méthode déterministe. La fonction estimée $\hat{f}$ permet de définir un estimateur de S avec peu de données. Pourtant, beaucoup d'itérations peuvent être nécessaires afin que $\hat{S}$ converge et soit proche de

S. De plus, la méthode est plus complexe que la précédente et nécessite un choix important de critères et de paramètres utiles à l'algorithme. La seconde méthode stochastique présente d'ailleurs les mêmes inconvénients.

2.2.4 Méthode MCMC

Cette méthode statistique consiste à modéliser les écarts entre le code de calculs et la réalité :

$$y = f(x) + \epsilon,$$

où ϵ représente l'erreur de modélisation, de loi normale centrée. La méthode est basée sur une approche bayésienne. On choisit une loi a priori $\pi_0(x)$ sur les entrées et on considère une forme paramétrique pour les erreurs, $p(\epsilon|x)$. Par la formule de Bayes, on obtient la loi a posteriori sur la valeur des entrées,

$$p(x|\epsilon) = \frac{p(\epsilon|x)\pi_0(x)}{p(\epsilon)}.$$

Les solutions du problème d'inversion sont les modes de la loi a posteriori : $x^* = \arg\max_x p(\epsilon|x)\pi_0(x)$. Le problème est que pour cela, il faut simuler suivant la loi a posteriori. Or cette loi n'est en général pas une loi standard. En effet, si la loi des écarts et la loi a priori sont des lois normales par exemple, la loi a posteriori n'est pas une loi normale.
Pour simuler suivant la loi a posteriori, on utilise la méthode MCMC (Gelfand and Smith, 1990). Pour cela, il existe deux principaux algorithmes : l'algorithme de Metropolis-Hastings et l'échantillonnage de Gibbs. Le principe est de considérer une approximation de la loi a posteriori, améliorée à chaque étape de l'algorithme. La chaîne de Markov est une suite dépendante de variables aléatoires ayant pour loi l'approximation de la loi a posteriori. Cette suite est utilisée pour évaluer des quantités d'intérêt sur la loi a posteriori, par méthode empirique. Ce sont ces quantités qui permettent d'améliorer itérativement la loi approximative. Dans l'algorithme de Metropolis-Hastings, un choix courant pour cette loi, appelée loi instrumentale, est de la considérer centrée sur le point en cours avec des mouvements locaux, souvent une loi de type gaussienne.
Cette méthode est applicable directement sur le code et permet de prendre en compte les incertitudes du système. Par contre, la méthode nécessite de faire des choix a priori et d'effectuer beaucoup de réglages. De plus, elle est coûteuse en temps de calculs et en pratique, il n'y a pas de critère d'arrêt bien défini à la méthode.

Face aux difficultés et aux inconvénients de toutes ces méthodes, nous en avons développé

deux nouvelles : la méthode MRM et la méthode COMET. Nous avons également à l'idée une troisième méthode dont nous avons traité uniquement une partie théorique mais qui, à terme, pourra fournir des résultats intéressants pour la résolution de problèmes inverses. Cette méthode fera l'objet du Chapitre V, qui pourra être omis en première lecture. Dans les sections ci-dessous, nous nous intéressons aux deux nouvelles méthodes d'inversion que nous avons développées. Ces méthodes ont été codées dans R et sont applicables à des problèmes inverses. Le but de ces méthodes est de fournir, sous certaines conditions de régularité de la fonction f, un grand nombre de combinaisons pour x telles que la variable d'intérêt se trouve dans un voisinage restreint de la valeur cible 0. Nous obtiendrons donc l'ensemble $V(tol) = \{x \in D \subset \mathbb{R}^d, |f(x)| \leq tol\}$ pour une tolérance tol choisie. Nous allons décrire dans un premier temps la méthode MRM puis la méthode COMET. Enfin, nous comparerons les deux sur des fonctions usuelles en dimension 2.

3 Nouvelle méthode : MRM (*Monotonous Reliability Method*)

A l'origine, la méthode MRM est une méthode qui a été développée par de Rocquigny et qui fut utilisée en fiabilité des structures. Le but de la méthode de de Rocquigny (Rocquigny, 2009) est d'estimer des probabilités de défaillance $p = \mathbb{P}(f(X) \leq 0)$, où X est un vecteur aléatoire et f une fonction monotone déterministe, généralement appelée fonction de performance. Le bénéfice de la monotonie de f a été exploité successivement dans (Rocquigny, 2009), (Limbourg et al., 2010) puis (Rajabalinejad et al., 2011). Associée à la continuité et à la simple connexité de la surface de défaillance $S = \{x, f(x) = 0\}$, la monotonie permet d'utiliser tout plan d'expériences pour proposer un encadrement déterministe de p. Ces bornes peuvent être calculées de manières exactes (Bousquet, 2012) quand les entrées du code sont supposées indépendantes et uniformes sur le domaine d'entrée de f. Mais on préfèrera en général une méthode de Monte-Carlo standard (Rocquigny, 2009), insensible à la dimension de l'espace des entrées. Lorsque ce plan d'expériences est choisi de façon stochastique, un estimateur statistique de p est proposé en parallèle des bornes. Il s'agit alors d'un estimateur de Monte-Carlo adaptatif (Rajabalinejad et al., 2011) pour lequel les tirages se font de manière uniforme dans une suite d'espaces imbriqués, construits par la monotonie. Certaines démonstrations et améliorations sont fournies dans (Bousquet, 2012).

Cette méthode a également été présentée dans (Popelin et al., 2012) lors du 18ème Congrès Lambda Mu, notamment les différentes étapes de la méthode MRM, dont nous reprendrons

certaines notations. Nous y trouvons également une comparaison avec les méthodes habituellement utilisées en fiabilité comme FORM, Monte-Carlo ou importance sampling via un exemple analytique. Enfin, les auteurs appliquent la méthode MRM à un code thermomécanique pour l'amorce de fissure.

Nous nous sommes inspirés de cette méthode pour approcher l'ensemble S. L'idée est que la monotonie de f, fonction que l'on étudie sur un ensemble fermé, nous permet, pour chaque point tiré dans cet ensemble, de réduire son volume jusqu'à ce que tous les points tirés dans l'ensemble restant soient proche de S, selon une tolérance choisie.

3.1 Principe de la méthode

Soit f une fonction déterministe dans le sens où, pour un même x, le calcul, répété plusieurs fois, de $f(x)$ donnera toujours le même résultat :

$$\begin{cases} \mathbb{U} \subset \mathbb{R}^d \to \mathbb{R} \\ x \mapsto f(x). \end{cases} \tag{IV.16}$$

Notations

$S = \{x \in \mathbb{U}, f(x) = 0\}$ est la surface recherchée,
$\mathbb{U} = [-1, 1]^d$ par normalisation des données d'entrée,
$\mathbb{U}^+ = \{x \in \mathbb{U}, f(x) > 0\}$,
$\mathbb{U}^- = \{x \in \mathbb{U}, f(x) < 0\}$.
On a alors $\mathbb{U} = S \cup \mathbb{U}^+ \cup \mathbb{U}^-$.

On suppose également que f est globalement monotone.

Définition 7 (Monotonie globale)**.** *On dit que f est globalement monotone sur $\mathbb{U}$ si $\forall i, \exists s^i \in \{-1, 1\}$ tel que*

$$\forall a \geq 0, \forall \vec{x} = (x_1, \ldots, x_i, \ldots, x_n) \in \mathbb{U}, f(x_1, \ldots, x_i + s^i a, \ldots, x_n) \leq f(x_1, \ldots, x_i, \ldots, x_n),$$

où s^i est le signe de la dépendance monotone. Ainsi, $s^i = 1$ (resp. $s^i = -1$) quand f est décroissante (resp. croissante) suivant la $i^{\text{ème}}$ composante x_i.

Hypothèse 1 Sans perte de généralité, on suppose que f est globalement croissante.

Hypothèse 2 S est continue et simplement connexe.

A chaque étape k, un plan d'expériences D_k est formé de part et d'autre de S. Composé de k points, il est séparé en deux ensembles :

$$D_k^+ = \{x \in D_k, f(x) > 0\},$$
$$D_k^- = \{x \in D_k, f(x) < 0\}.$$

Comme f est globalement croissante, si $x \in D_k^+$, alors tout point t tel que $x \preceq t$ est également dans D_k^+. De même, si $x \in D_k^-$, alors tout point t tel que $t \preceq x$ est également dans D_k^-. La relation $\preceq$ désigne un ordre partiel que l'on définit de la manière suivante :

Définition 8 (Ordre partiel dans $\mathbb{R}^d$)**.** *On dit qu'un élément est inférieur à un autre suivant l'ordre partiel $\preceq$ si toutes les coordonnées du premier sont inférieures à celles du deuxième. On peut l'écrire :*

$$\forall (x, y) \in \mathbb{U}^2, (x \preceq y) \Leftrightarrow (\forall k \in [\![1, d]\!], x_k \leq y_k).$$

On définit alors les trois ensembles suivants :

$$\mathbb{U}_k^+ = \{x \in \mathbb{U} : y \preceq x, y \in D_k, f(x) > 0\},$$
$$\mathbb{U}_k^- = \{x \in \mathbb{U} : x \preceq y, y \in D_k, f(x) < 0\},$$
$$\mathbb{U}_k = \mathbb{U} \setminus \{\mathbb{U}_k^+ \cup \mathbb{U}_k^-\}.$$

A chaque itération, un « hyper-pavé » de taille d est éliminé, il correspond à la simulation d'un point sur le domaine. Ce pavé se trouve soit dans le coin nord-est du domaine de définition $\mathbb{U}$ dans le cas d'un point situé dans la zone positive $\mathbb{U}_k^+$, soit dans le coin sud-ouest dans le cas d'un point situé dans la zone négative $\mathbb{U}_k^-$. La Figure IV.12 illustre la méthode dans le cas $d = 2$, c'est-à-dire que S, l'ensemble des points x pour lesquels $f(x) = 0$, est une courbe, représentée en rouge sur la figure. On observe les pavés associés à 10 points simulés dans le domaine initial, $[-1, 1]^2$. Comme f est globalement croissante, la partie du domaine située au-dessus de S forme $\mathbb{U}^+$, en-dessous, il s'agit de $\mathbb{U}^-$. Six points ont été simulés dans $\mathbb{U}^+$ (numérotés de 1 à 6), quatre dans $\mathbb{U}^-$ (numérotés de 7 à 10). Pour chaque point, son pavé associé a été tracé (rectangles bleus et roses). On voit que les points 2 à 5 se situent dans le pavé associé au point 1. Chacun de ces points est supérieur, au sens de l'ordre partiel, au point 1. L'image de ces 4 points par f étant alors supérieure à celle du point 1, il est donc inutile en pratique d'évaluer f en ces points connaissant l'image du point 1.

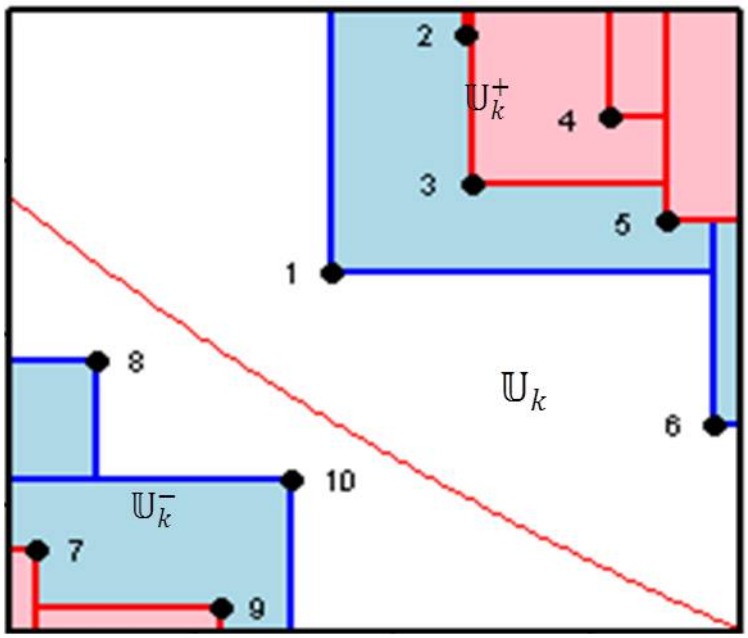

Figure IV.12 – Illustration de la méthode MRM en deux dimensions

Les unions des pavés constitués dans $\mathbb{U}^+$ et dans $\mathbb{U}^-$ forment respectivement $\mathbb{U}_k^+$ et $\mathbb{U}_k^-$, les ensembles de l'étape k. Notons que les points 1 et 6 sont appelés les sommets de $\mathbb{U}_k^+$. De même, les points 8 et 10 sont les sommets de $\mathbb{U}_k^-$. L'espace blanc restant forme $\mathbb{U}_k$. À l'étape suivante, les points seront simulés dans $\mathbb{U}_k$. C'est ainsi que l'on tire profit de l'hypothèse de monotonie. L'idée de la méthode est de réduire $\mathbb{U}_k$ à chaque étape de l'algorithme, jusqu'à ce qu'il soit très proche de S. La démonstration de convergence de cette méthode est fournie en Annexe C.
La difficulté est de trouver les façons de tirer le point suivant dans le domaine $\mathbb{U}_k$. Ceci fait l'objet des prochaines sections.

3.2 Algorithme de base

La base de toutes les variantes que nous proposons consiste en un algorithme que l'on peut découper en trois parties. A l'étape $k \geq 1$, il s'agit de

1. choisir un plan d'expériences dans le domaine inexploré $\mathbb{U}_{k-1}$ obtenu à l'étape précédente. Le nombre de points du plan peut changer suivant la variante de la méthode,
2. évaluer $f(.)$ sur les points du plan d'expériences,

3. mettre à jour les sous-espaces

$$\mathbb{U}_k^+ = \mathbb{U}_{k-1}^+ \cup \{x \in \mathbb{U} | \exists x_k, f(x_k) > 0, x \succeq x_k\},$$
$$\mathbb{U}_k^- = \mathbb{U}_{k-1}^- \cup \{x \in \mathbb{U} | \exists x_k, f(x_k) < 0, x \preceq x_k\},$$
$$\mathbb{U}_k = \mathbb{U} \setminus (\mathbb{U}_k^+ \cup \mathbb{U}_k^-).$$

Remarquons que dans son utilisation pour l'estimation d'une probabilité de défaillance, cet algorithme comporte une étape supplémentaire consistant à mettre à jour les bornes de cette probabilité.

Après avoir traité le problème de l'initialisation de l'algorithme, nous proposerons plusieurs variantes qui sont des méthodes différentes pour choisir, à chaque itération, le plan d'expériences de l'étape 1 de l'algorithme.

3.3 Initialisation

À la Figure IV.12, on voit bien qu'une initialisation consistant à simuler aléatoirement un grand nombre de points n'est pas très efficace. En effet, certains points risquent d'être inutiles (comme les points 2 à 5 ou encore 7 et 9). L'idée est également d'éliminer une grande partie du domaine de part et d'autre de S. En simulant deux points aléatoirement sur $\mathbb{U}$, on risquerait d'avoir des points éloignés de S ou situés dans la même zone ($\mathbb{U}^+$ ou $\mathbb{U}^-$). L'initialisation par dichotomie permet d'obtenir deux points proches mais situés de part et d'autre de S, ce qui permet d'éliminer une grande partie du domaine. Cela se fait sur la diagonale $S - O \to N - E$ du domaine de variation. Le milieu de cette diagonale est évalué (on calcule l'image de ce point par f). S'il se trouve dans la zone positive $\mathbb{U}^+$, alors le prochain point évalué sera au quart de la diagonale, sinon, il sera aux trois quarts. Le critère d'arrêt porte sur l'existence de deux points situés de part et d'autre de S, dans une zone de tolérance choisie. On parlera de points solutions. Notons x^+ et x^- ces deux points et tol le seuil de tolérance choisi. L'étape s'arrête lorsque $|f(x^+)| < tol$ et $|f(x^-)| < tol$.
Cette initialisation par dichotomie est illustrée dans le cas bidimensionnel à la Figure IV.13, pour un seuil fixé à 10^{-2}. La courbe rouge représente $S = X \in \mathbb{R}^2, f(X) = 0$. La droite en pointillés est la diagonale $S - O \to N - E$ sur laquelle on effectue la dichotomie. Les points noirs sont les points en lesquels f a été successivement évaluée. Les carrés bleus sont les parties du domaine éliminées au fur et à mesure du processus dichotomique. Dans ce cas précis, 8 simulations ont été nécessaires pour obtenir deux points dans la zone de tolérance $[-10^{-2}, 10^{-2}]$.

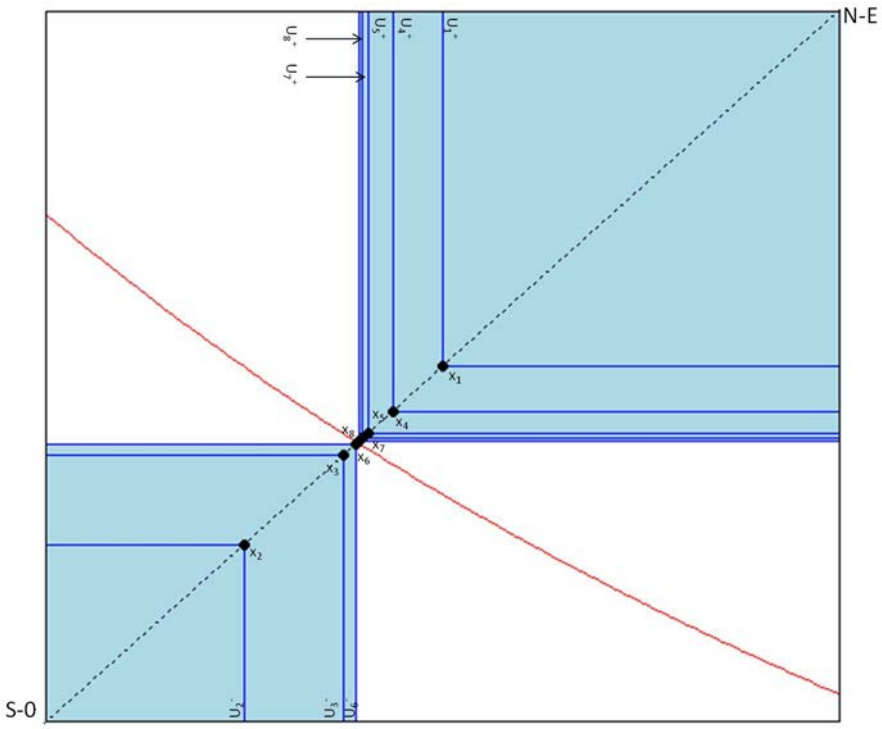

Figure IV.13 – Illustration de la méthode sur la diagonale S-O→N-E en deux dimensions pour une tolérance à 10^{-2}

Ce nombre dépend de la dimension, de la fonction étudiée et de la tolérance choisie. L'évolution du nombre moyen de simulations nécessaires en fonction du seuil choisi par exemple, est inversement logarithmique, comme le montre la Figure IV.14. Il est à noter que la tolérance a été prise entre 10^{-3} et 1 avec un pas de 5.10^{-3}. Ce que nous observons à la Figure IV.14 est valable dans le cas précis de l'étude de cette fonction mais n'est pas exactement identique pour toutes les fonctions même si c'est assez similaire puisque les fonctions étudiées sont d'un type particulier, les fonctions monotones.

En dimension 3, il faut entre 0 et 2 points supplémentaires par rapport à la dimension 2 pour une tolérance donnée. La décroissance du nombre d'évaluations nécessaires est en $-log(tolérance)$. Il faut donc diminuer la tolérance de départ si le nombre d'évaluations de f est limité a priori. On note N_0 le nombre de points évalués lors de l'initialisation. On pose alors $\mathbb{U}_0^+$ et $\mathbb{U}_0^-$ les ensembles de base, obtenus après N_0 simulations et qui vont être utilisés à l'étape 1 de l'algorithme.

Cette méthode d'initialisation est aussi conseillée par (Bousquet, 2012). En effet, cela permettrait de réduire de manière significative la largeur de l'intervalle constituant les bornes de la probabilité de défaillance p.

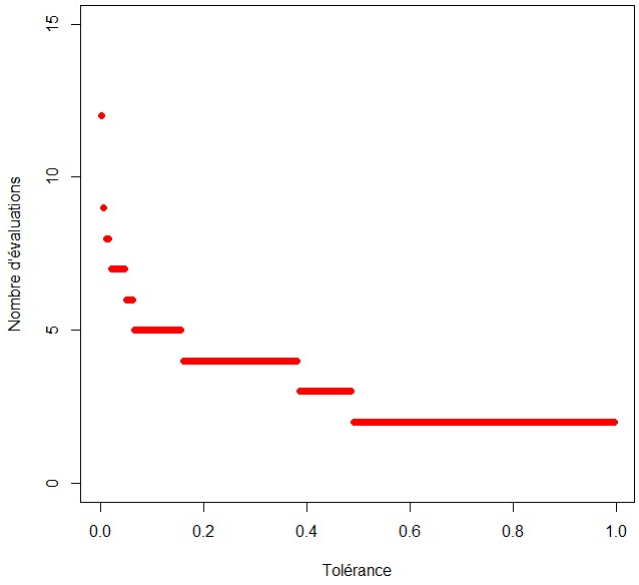

Figure IV.14 – Évolution du nombre d'évaluations en fonction de la tolérance

3.4 Algorithme déterministe : poursuite par dichotomie

Pour l'algorithme de dichotomie, il suffit d'obtenir un point dans la zone de tolérance. L'initialisation ne nécessite donc que 6 simulations. Les points solutions suivants seront recherchés sur des droites parallèles à cette diagonale. Le choix des droites étudiées se fera de manière dichotomique sur la diagonale $S-E \to N-O$. À l'étape d'initialisation, on a obtenu un point sur la médiatrice de la diagonale $S-E \to N-O$. À l'étape 1 de l'algorithme, on cherchera les points solutions sur les perpendiculaires à $S-E \to N-O$, coupant cette diagonale à son quart et ses trois quarts. Nous aurons donc 2 points solutions supplémentaires. À l'étape 2, on étudiera le cas des perpendiculaires passant par les huitièmes de la diagonale. Ceci fournira 4 points solutions. Et ainsi de suite. Toutes les droites étudiées sont d'équation $y = x + b$, comme le montre la Figure IV.15. La diagonale $S-E \to N-O$ est représentée par des pointillés. La droite noire est la première à être étudiée (initialisation), son ordonnée à l'origine est $b = 0$. Les droites rouges, d'ordonnées à l'origine $b = 1, -1$, sont étudiées à l'étape 1. Les droites vertes, d'ordonnées à l'origine $b = -3/2, -1/2, 1/2, 3/2$ sont étudiées à l'étape 2. Les points noirs sont les intersections entre chacune de ces droites et la diagonale $N-O \to S-E$. Ce sont donc les milieux de chaque droite et par conséquent les points de départ de chaque étape de la dichotomie. Les points visés sont les gros points aux intersections entre ces droites et la courbe solution S. On remarque sur la Figure IV.15 que certaines droites ne coupent pas S dans le domaine considéré, il n'y aura donc pas de solution pour ces droites. En règle générale,

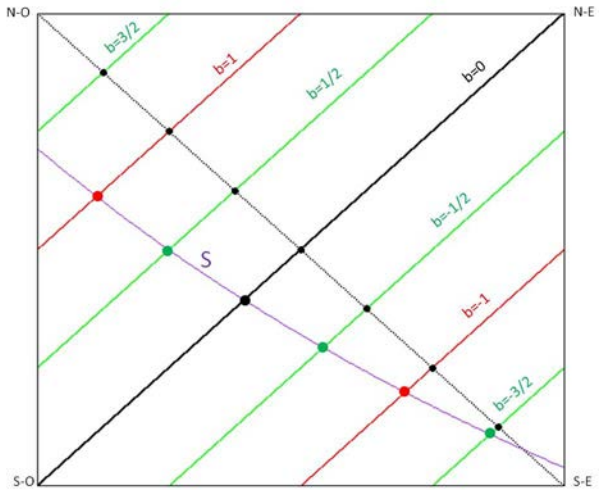

Figure IV.15 – Méthode MRM par dichotonmie

il y a une solution sur une droite Δ si

$$\begin{cases} \inf_{x\in[-1,1]\cap\Delta} f(x) < 0, \\ \sup_{x\in[-1,1]\cap\Delta} f(x) > 0. \end{cases}$$

Ici, en trois itérations (l'initialisation compte comme une itération), sept droites ont donné six points solutions. En k itérations, on aura $2^k - 1$ droites qui pourront donner au plus $2^k - 1$ points solutions. Si on veut au moins 1000 points solutions, il faudra donc faire au moins neuf itérations (peut-être plus si beaucoup de droites n'ont pas de résultat). Le choix de k constitue le critère d'arrêt de l'algorithme.

Cette méthode est applicable en toute dimension grâce à la monotonie globale. En effet, toutes les droites parallèles à la diagonale $[(-1)^k, (1)^k]$, où $(1)^k = \underbrace{(1,\ldots,1)}_{k}$ et $(-1)^k = \underbrace{(-1,\ldots,-1)}_{k}$ sont des coins de l'hyper-pavé et coupent la surface réponse en au plus un point. La difficulté est de trouver les équations des droites parallèles à la diagonale en grande dimension.
Le principal défaut de cette méthode déterministe est qu'elle demande beaucoup d'appels à la fonction f, ce qui peut être gênant dans le cas où l'évaluation de f est coûteuse ou que le nombre d'évaluations autorisé est limité.

La suite de l'algorithme se fera donc plutôt de manière stochastique. A chaque itération $k \geq 1$, il va s'agir de tirer aléatoirement un ensemble de points selon une certaine loi sur $\mathbb{U}_{k-1}$. Le

problème est que le domaine $\mathbb{U}_{k-1}$ n'est pas régulier. Les choix du nombre de points tirés à chaque itération et de la loi suivant laquelle ils sont tirés peuvent donner différents algorithmes selon des stratégies de tirage différentes. Ceci est présenté dans les sections suivantes.

3.5 Algorithme stochastique intuitif : Monte-Carlo adaptatif

La méthode la plus naturelle est basée sur des tirages de Monte-Carlo avec une loi uniforme sur l'espace non exploré $\mathbb{U}_k$. Comme il est difficile de tirer aléatoirement sur un ensemble différent de I^d où I est un intervalle, l'idée est d'adapter le Monte-Carlo au domaine inexploré. Nous ferons donc du Monte-Carlo adaptatif qui consiste à simuler un point uniformément sur le domaine initial et à l'évaluer seulement s'il est dans le domaine inexploré. S'il n'est pas dans $\mathbb{U}_k$, on retire jusqu'à ce que cet évènement ait lieu. La monotonie constitue le point fort dans cette méthode car, grâce à cette propriété, un point peut être rejeté sans avoir été évalué. Avec cette stratégie, le nombre d'éléments de l'échantillon évalué n'augmente que de 1 à chaque itération. Appelée « one-step ahead strategy », elle a été favorisée dans (Bousquet, 2012) et (Rajabalinejad et al., 2011).

La particularité de cet algorithme est que, à chaque étape, nous ne gardons que les sommets de $\mathbb{U}_k^+$ et $\mathbb{U}_k^-$. Ceci fait que l'échantillon à chaque itération ne comporte que les meilleurs points (au sens de leur proximité à S) simulés au cours de l'algorithme.
Plusieurs critères d'arrêt peuvent être envisagés pour cette méthode. Dans tous les cas, il faut fixer une tolérance autour de la solution. Trois critères sont proposés ici :

- le nombre de points situés dans la zone de tolérance est supérieur à un certain seuil N_f,
- tous les points de l'échantillon sont dans la zone de tolérance,
- une très grande majorité (95% par exemple) des points se trouve dans la zone de tolérance si le choix précédent est trop long à obtenir.

Cette méthode de Monte-Carlo adaptatif présente un certain nombre d'avantages et d'inconvénients. Ses principaux avantages sont la possibilité de l'appliquer en toutes dimensions, la simplicité de compréhension et de mise en application, son efficacité avec l'évaluation des points uniquement dans le domaine inexploré. Ce dernier point, dû à la monotonie de la fonction, fait que peu d'appels à la fonction sont nécessaires pour obtenir des points solutions.
Parmi ses principaux inconvénients, on a la dépendance de la rapidité et de l'efficacité de l'algorithme à la dimension du problème et à la tolérance choisie. Ainsi, plus la dimension est importante, plus le domaine à couvrir est grand, plus l'algorithme a de difficultés à trouver des points solutions. De la même façon, plus la tolérance est petite, plus le domaine inexploré est

petit, plus il est difficile de tirer aléatoirement un point dans cette zone (la probabilité tend même vers 0 quand k tend vers l'infini). La vitesse de convergence de l'algorithme diminue donc avec la tolérance.

3.6 Méthode semi-stochastique : la méthode des segments

L'idée de la méthode est de tirer davantage de profits des points déjà simulés. En effet, lorsqu'on dispose de deux points, un dans la zone positive, un dans la zone négative, il apparaît évident que tout point situé sur le segment formé par ces deux points sera plus proche de S. Instinctivement, on aurait tendance à vouloir relier tous les sommets de $\mathbb{U}_k^+$ à tous ceux de $\mathbb{U}_k^-$, que l'on note respectivement S_k^+ et S_k^-. Or, cela risque de ne pas être très efficace car de nombreuses droites vont se chevaucher, les nouveaux points de l'échantillon seront donc très proches, ce qui fournirait de l'information redondante. Une alternative consiste à relier les points selon des groupes formés par la méthode des k-means. Algorithme de clustering par partitionnement, la méthode des k-means permet de construire des collections d'objets par un apprentissage non supervisé. Le but est de constituer des groupes homogènes et bien séparés les uns des autres. Le lecteur intéressé pourra se référer à (MacQueen et al., 1967) ou (Hartigan, 1975). Le nombre de groupes $N_g = min(card(S_k^+), card(S_k^-))$ est le nombre de sommets de l'ensemble qui en comporte le moins. Dans cet ensemble, il n'y aura qu'un point par groupe. Dans l'ensemble restant, la méthode des k-means est appliquée pour répartir les sommets en N_g groupes. Chaque point du groupe i sera relié au point i de l'autre ensemble, comme le montre la Figure IV.16. Ici, l'espace $\mathbb{U}_k^-$ comporte moins de sommets que $\mathbb{U}_k^+$ qui ont donc été répartis en 3 groupes suivant la méthode des k-means. Les points verts sont les milieux des segments tracés entre les points de mêmes groupes.
Le réflexe est de prendre effectivement le milieu des segments comme nouveau point de l'échantillon. Or, il est possible de se rapprocher de S beaucoup plus rapidement en prenant en compte la valeur de f en chacune des extrémités des segments étudiés. Au lieu de prendre le milieu de chaque segment, nous considérerons plutôt le barycentre inversé dont les pondérations dépendent des valeurs de f aux extrémités.
Considérons les sommets A et B de coordonnées $(x_A^{(i)})_{i=1,\dots,n}$ et $(x_B^{(i)})_{i=1,\dots,n}$ dans $[-1,1]^n$. La fonction f vaut respectivement f_A et f_B en ces points. Les coordonnées du nouveau point M, barycentre inversé de A et B, sont définis ainsi :

$$\forall i \in [\![1;n]\!], x_M^{(i)} = \frac{|f_B|}{|f_B - f_A|} x_A^{(i)} + \frac{|f_A|}{|f_B - f_A|} x_B^{(i)}.$$

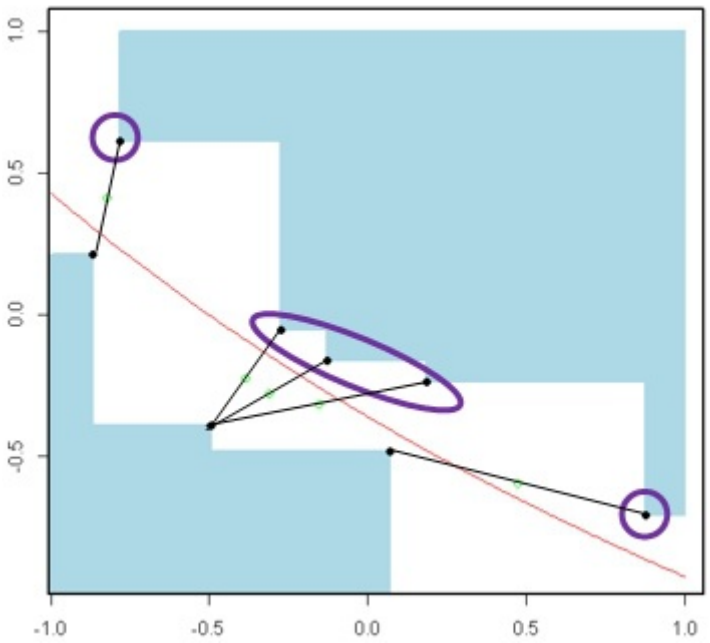

Figure IV.16 – Illustration de la méthode de k-means lors d'une itération de l'algorithme MRM

On parle de barycentre inversé car plus l'image du point par f est petite en valeur absolue (plus le point est proche de S), plus son poids sera important dans le calcul du barycentre. Les poids de A et B sont inversés par rapport à un barycentre standard.

Pour que cette méthode soit efficace, il est clair qu'il faudra un grand nombre de points de part et d'autre de S, assez bien répartis sur le domaine. En effet, cette méthode aura tendance à accentuer la proximité entre les points de l'échantillon et pourrait laisser des zones de côté. Il est donc impossible de l'utiliser directement avec l'initialisation que nous avons présentée auparavant. Deux solutions s'offrent alors : soit on simule uniformément un échantillon sur l'ensemble de départ $\mathbb{U}$ au risque d'avoir un grand nombre de points inutiles car inclus dans la zone de rejet d'un autre point, soit on la couple à la méthode de Monte-Carlo adaptatif afin de pallier le problème de la probabilité très faible de tirer dans l'espace inexploré $\mathbb{U}_k$ quand k est grand. Dans ce cas, il faut définir le moment où le Monte-Carlo adaptatif s'arrête (nombre fixé d'évaluations de f atteint, pourcentage de points tirés dans la zone inexplorée, temps de l'algorithme, etc).

3.7 Algorithme particulier pour le cas bidimensionnel : méthode des rectangles

Une autre alternative pour répondre aux différents problèmes de la méthode de Monte-Carlo adaptatif est évidemment de tirer les points directement dans le domaine inexploré et non plus dans le domaine initial. Seulement, les tirages aléatoires se font sur des hyper-pavés, formes régulières dont on connaît les bornes sur chaque dimension. L'idée est donc, vue la configuration de l'espace inexploré $\mathbb{U}_k$ à l'étape k (cf. Figure IV.12), de le découper en hyper-cubes. En dimension 2, le partage en rectangle est relativement simple mais à partir de la dimension 3, cela devient bien plus compliqué. Nous n'appliquerons donc cette méthode que dans le cas bidimensionnel.

Après l'initialisation par dichotomie, on dispose d'un domaine inexploré composé de deux rectangles de même aire comme on peut le voir à la Figure IV.13. Dans chacun de ces rectangles, on simule un certain nombre de points (la discussion concernant le choix de ce nombre sera faite plus loin). En chacun de ces points, on trace une droite verticale (tracer une droite horizontale reviendrait au même). Pour une étape où nous disposons de k points, il y aura $k+1$ rectangles comme le montre la Figure IV.17. Dans le cas de points de même abscisse, les rectangles sont confondus. Le découpage de la zone inexplorée en rectangles fournit deux types de zones : les zones blanches sont les zones rejetées, les zones roses et grises sont les rectangles dans lesquels on va simuler les points de l'étape suivante. Concernant le choix du nombre de points dans chaque rectangle, il y en a 3 possibles :

- simuler un point par rectangle,
- prendre le milieu de chaque rectangle,
- simuler un nombre de points n_i proportionnel à l'aire A_i du rectangle vis-à-vis de l'aire A_{tot} du domaine inexploré, qui n'est autre que la somme des aires des différents rectangles. Il est important pour ce choix de faire en sorte d'avoir au moins un point dans chaque rectangle ou presque. Par exemple, on peut choisir que les rectangles pour lesquels $A_i \geq \frac{A_{tot}}{10}$ sont de taille suffisante pour apporter de l'information, on y simulera au moins un point. Dans ce cas, $n_i = \frac{10A_i}{A_{tot}}$.

Le critère d'arrêt de la méthode porte sur la distance entre les sommets NE et SO (du fait de la croissance globale) de chaque rectangle et S. Ainsi, dès que la valeur en f d'un de ces sommets est, en valeur absolue, inférieure à une tolérance choisie, alors l'algorithme s'arrête.

Cette méthode est rapide puisqu'elle permet de simuler des points directement dans la zone

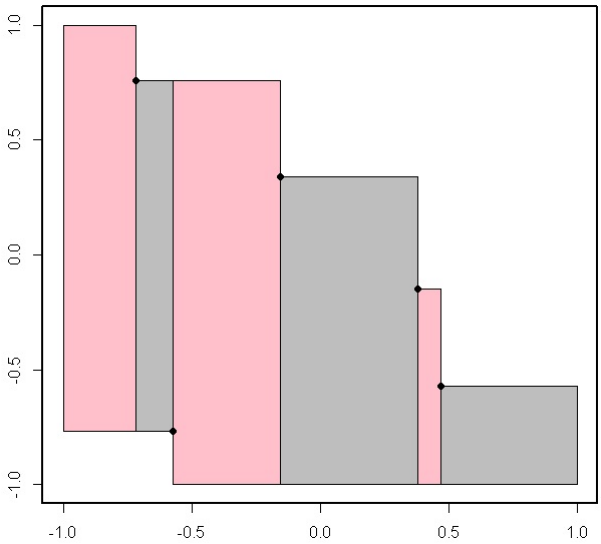

Figure IV.17 – Utilisation d'un découpage par rectangle pour la méthode MRM

inexplorée. D'ailleurs, une fois que l'algorithme a convergé, on dispose d'un certain nombre de points solutions mais on peut en obtenir une infinité en simulant dans les rectangles restants. En effet, étant donnée la nature du critère d'arrêt, tout point simulé dans un de ces rectangles sera dans la zone de tolérance. En outre, il ne sera pas nécessaire d'évaluer ces points en f. Enfin, la méthode exige assez peu d'appels à la fonction f.
Par contre, elle est assez fastidieuse à coder donc difficilement généralisable en dimension supérieure. Ce problème est d'autant plus important que les rectangles en dimension 2 deviennent des hyper-pavés en dimensions supérieures. Leur définition devient donc difficile avec l'augmentation de la dimension.

D'autres améliorations de la méthode initiale sont proposées dans (Moutoussamy, 2015). La thèse présente une version de MRM par tirage d'importance adaptatif et une version de MRM pour l'estimation de quantiles. Les méthodes développées sont testées sur des exemples jouets.

3.8 Traitement de la condition de monotonie

Plusieurs voies sont envisagées afin de considérer des cas plus généraux de la méthode. Il est par exemple possible d'alléger l'hypothèse de monotonie de la fonction. En effet, nous pouvons nous contenter d'une monotonie par morceaux, ce qui exige un travail préliminaire sur la fonction. Si f est différentiable en chacune de ses variables, il est possible de calculer ses dérivées

partielles du premier ordre.

Prenons l'exemple où f est une fonction de deux variables x_1 et x_2. Les deux dérivées partielles du premier ordre sont notées $\frac{\partial f}{\partial x_1}$ et $\frac{\partial f}{\partial x_2}$. L'étude du signe de chacune de ces deux fonctions nous donne les variations de f en chacune de ses variables. L'idée de la méthode que nous proposons est de repérer les changements de signe des dérivées partielles. Pour cela, nous quadrillons assez finement le domaine de variation $[-1, 1]^2$. Les points du quadrillage sont triés suivant le signe des dérivées partielles. Dans notre exemple, il y a quatre groupes : ce sont les ensembles des couples (x_1, x_2) tels que

1. $\frac{\partial f}{\partial x_1} \leq 0$ et $\frac{\partial f}{\partial x_2} > 0$ pour le groupe 1,
2. $\frac{\partial f}{\partial x_1} > 0$ et $\frac{\partial f}{\partial x_2} > 0$ pour le groupe 2,
3. $\frac{\partial f}{\partial x_1} \leq 0$ et $\frac{\partial f}{\partial x_2} \leq 0$ pour le groupe 3,
4. $\frac{\partial f}{\partial x_1} > 0$ et $\frac{\partial f}{\partial x_2} \leq 0$ pour le groupe 4.

Ces groupes peuvent être continus ou discontinus, et de forme plus ou moins régulière. La Figure IV.18 représente deux exemples. Le premier est le résultat de l'étude pour $f(x_1, x_2) = x_1^2 + x_2^2 - 0.25$, le second pour $f(x_1, x_2) = x_1^3 - 3x_2^2 - x_1 + 2x_1x_2$. Le groupe 1 est en violet,

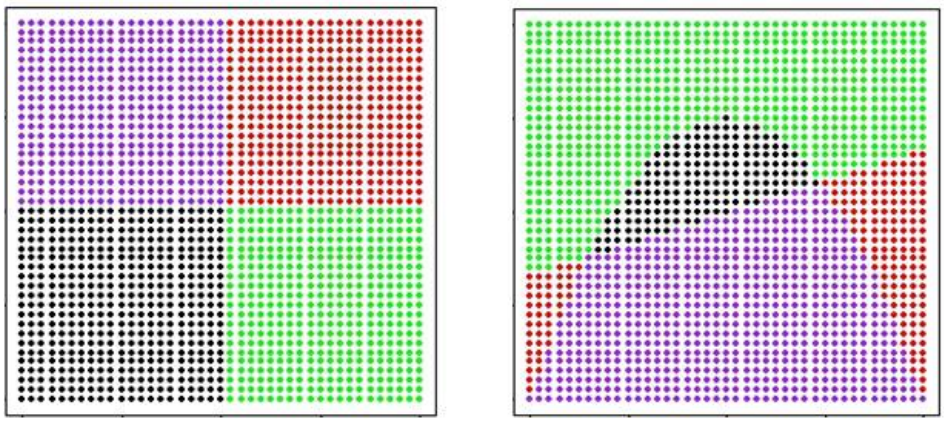

Figure IV.18 – Deux exemples de regroupement des points du quadrillage par groupes de monotonie

le groupe 2 en rouge, le groupe 3 en noir et le groupe 4 en vert. Dans le premier exemple, les groupes sont continus et de forme régulière (leurs bords sont droits). Il suffit donc de découper le domaine de variation suivant ces limites et appliquer la MRM sur chaque sous-domaine. Dans le second exemple, le groupe 2 est discontinu et les groupes sont irréguliers. Les groupes sont bien séparés mais il est impossible d'appliquer la méthode MRM sur des ensembles dont les limites ne sont pas explicitement connues. En effet, la méthode MRM simule uniformément des points sur des domaines rectangles. Une idée est d'estimer la loi des points de chaque zone puis

de simuler les points de la MRM suivant cette loi.
Bien sûr, plus la fonction f comporte de variables, plus il y a de groupes et plus la méthode de regroupement par monotonie est compliquée. En effet, si f a n variables, alors il y a 2^n groupes de monotonie possibles.

Une autre voie de généralisation consiste à faire abstraction de la monotonie de la fonction. Dans la méthode que nous proposons, nous considérons les groupes de monotonie (on parlera ici de zones) avec la méthode d'élimination de coins du domaine par la MRM. Pour chaque point simulé dans le domaine, il y a un pavé potentiellement éliminé qui lui est associé.
Reprenons le cas bidimensionnel $f(x_1, x_2) = x_1^2 + x_2^2 - 0,25$. Dans ce cas, la solution de $f(x_1, x_2) = 0$ est le cercle de centre $(0, 0)$ et de rayon 0.5. Soit $X = (x_1, x_2)$ un point simulé dans le domaine de variation $[-1, 1]^2$. On a quatre cas de figure :

1. X est dans la zone 1 et $f(X) > 0$ ou X est dans la zone 4 et $f(X) < 0$: le pavé se situe dans le coin en haut à gauche.
2. X est dans la zone 1 et $f(X) < 0$ ou X est dans la zone 4 et $f(X) > 0$: le pavé se situe dans le coin en bas à droite.
3. X est dans la zone 2 et $f(X) > 0$ ou X est dans la zone 3 et $f(X) < 0$: le pavé se situe dans le coin en haut à droite.
4. X est dans la zone 2 et $f(X) < 0$ ou X est dans la zone 3 et $f(X) > 0$: le pavé se situe dans le coin en bas à gauche.

L'idée de la méthode est de simuler un certain nombre de points dans le pavé associé au point étudié. S'il y a un changement de signe au niveau de la fonction f ou de ses dérivées partielles du premier ordre, alors on ne prend pas en compte le point. Cela permettrait d'approcher le cercle par l'extérieur. Cette méthode est facilement généralisable à d'autres fonctions mais devient compliquée en plus grande dimension. En effet, en dimension n, on aurait 2^n zones et autant de coins d'élimination possibles. La difficulté réside surtout dans la complexité du code à développer.
Dans le cas du cercle, nous avons choisi de simuler 20 points dans chaque pavé associé à un point sélectionné. Si parmi ces 20 points, il n'y a pas de changement de signe de f ou de ses dérivées premières, alors on garde le point et on élimine le pavé qui lui est associé. Le point suivant est tiré sur le domaine restant par Monte-Carlo adaptatif. Il faut entre 2 et 3 minutes pour obtenir plus de 400 points dans un voisinage à 10^{-1} du cercle, résultat de l'équation $f(x, y) = 0$. Afin d'accélérer la procédure, nous pourrions par exemple remplacer le Monte-Carlo adaptatif par la méthode des rectangles développées précédemment.

Face aux difficultés et aux restrictions de la méthode MRM, nous avons développé une nouvelle méthode, la méthode COMET.

4 Nouvelle méthode en deux variantes : SAFIP et COMET

Les deux variantes permettent de résoudre $f(x) = 0$ où $f : D \subset \mathbb{R}^d \to \mathbb{R}$, $d > 1$ avec aucune hypothèse forte sur la fonction f ni sur l'ensemble des solutions $S = \{x : f(x) = 0\}$.
Comme pour la méthode MRM, leur but est de trouver des points dans une zone de tolérance faible autour de S. En gardant les notations de la section précédente, nous cherchons des points dans l'ensemble $\tilde{S} = \{x : |f(x)| \leq tol\}$, où tol est une tolérance (précision) choisie.
SAFIP (A Streaming Algorithm for Inverse Problems) est une méthode basée sur la construction de chaînes décroissantes en f. La convergence de cette méthode a été démontrée, il s'agit de l'article présenté dans la section suivante.
COMET (Constrained Optimization Method for Target achievement) est une méthode inspirée de l'algorithme tabou et de l'estimation par noyau. Son nom vient du fait que tout problème inverse peut être considéré comme un problème d'optimisation par atteinte de cible et que c'est cette méthode qui sera ensuite utilisée sur le cas test industriel qui est un problème sous contraintes.
Les deux variantes de cette nouvelle méthode sont testées sur des fonctions plus ou moins régulières, habituellement utilisées pour tester les méthodes d'optimisation.

4.1 Description de la première version de l'algorithme : article « SAFIP : A Streaming Algorithm for Inverse Problems »

Abstract This paper presents a new algorithm which aims at the resolution of inverse problems of the form $f(x) = 0$, for $x \in \mathbb{R}^d$ and f an arbitrary function with mild regularity condition. The set of solutions S may be infinite. This algorithm produces a good coverage of S, with a limited number of evaluations of the function f. It is therefore appropriate for complex problems where those evaluations are costly. Various examples are presented, with d varying from 2 to 10. Proofs of convergence and of coverage of S are presented.

4.1.1 Introduction

The scope of this paper Assume that we are given a bounded and closed domain $D \subset \mathbb{R}^d$, and a continuous real-valued function f defined on D.
The aim of this paper is to present an algorithm for the solution of the problem

$$S = \{x \in D : f(x) = 0\}, \tag{IV.17}$$

assuming $S \neq \emptyset$.
Such problems have been extensively handled over the years ; see Nakamura and Potthast (2015). The difficulty which we are confronted to lies in three main points :

1. the set S may contain many points, even be infinite,
2. the function f might be quite costly for example when defined by a simulation device,
3. the function f may be quite irregular ; we will assume mild regularity in the neighborhood of any point in S, only.

We also provide a two-fold proof for the convergence of this algorithm, namely we first prove that any resulting sequence of points in D converges to some point in S, and secondly that any point x in S is reached asymptotically by some "good" sequence, which is a sequence starting in a suitable neighborhood of x. As usually done in random search techniques, the starting points will be defined through random sampling in D.

Bibliographic outlook Most approaches to Problem (IV.17) extensively use analytic properties of the function f ; dichotomy, false position, Newton, conjugate gradient, etc (see Süli and Mayers (2003)) handle so called well-posed problems, when the equation $f(x) = 0$, for $x \in \mathbb{R}$ and f a real-valued function, has a unique solution. The case where f is defined as a mapping from $\mathbb{R}^d$ to $\mathbb{R}^k$ with $d \leq k$ is treated by singular value decomposition (see Golub and Loan (2012)), which also solves well-posed problems.
The ill-posed problems which we consider, namely the case where Problem (IV.17) has multiple solutions, is usually handled through regularization techniques, which aim at transposing (IV.17) into a well-posed problem. This procedure produces a partial solution to (IV.17) under appropriate knowledge on the function f (see Tikhonov et al. (2013)). All these techniques are out of the concern of the present work, where all solutions of f are looked for, with minimal assumption on f. We briefly present four methods, which constitute the environment of our proposal.

Local multi-start optimization, a deterministic approach Looking for the value of x such that $f(x) = 0$, consider the function $|f|$; minimizing $|f|$ indeed produces the set S.
First we choose a local optimization technique (Newton-Raphson for example). Then consider a design, which is a grid of initial points for the local optimization. From any of those, the sequence of iterations of the local optimization algorithm may produce a limit solution in S. Obviously stationary points not in S may be produced. The initial design is of utmost importance and the method may be unstable in this respect. Furthermore the method may be very costly due to the numerous evaluations of f. A general reference for those methods is György and Kocsis (2011).

A grid search, deterministic approach This method produces a sequence of grids in D. Given an initial regular grid, the function f is evaluated on each of its points. Points where f is close to 0 are selected and the grid is updated and refined in the neighborhood of those points. This method has been proposed by Miller (2005). A serious drawback lies in its cost, when the dimension of D corresponds to real life cases. Furthermore, the stopping rule of such algorithms does not guarantee a uniform approximation of S.

A Monte Carlo Markov Chain technique We assume that the function f is written as $f(x) = g(x) + \epsilon$. f is then a model for the real function g with an error ϵ due to modelling. For example, g is a physical model and f a computer-based formula for g. We estimate $S = \{x : g(x) = 0\}$. We choose a prior distribution $\Pi_0(x)$ on $\mathcal{X}$ and a parametric form for the distribution of ϵ, $p(\epsilon|x)$, for fixed x. By Bayes formula, the a posteriori distribution of x given ϵ is given by

$$\frac{p(\epsilon|x)\Pi_0(x)}{\int p(\epsilon|x)\Pi_0(x)dx}. \tag{IV.18}$$

The maximum probability principle provides stochastic solutions of $g(x) = 0$ as the maximum of (IV.18) upon x, given the prior Π_0.
In turn it can be proved that, whenever $\Pi_0(x) = \mathcal{N}(x_0, \sigma_0^2)$ the Gaussian distribution with mean x_0 and variance σ_0^2, for some $x_0 \in D$ and $\sigma_0^2 > 0$, solutions x^* of (IV.18) can be written as

$$x^* := argmin_{x \in D} ||y - g(x)|| + \frac{\sigma_\epsilon^2}{\sigma_0^2}||x - x_0||^2, \tag{IV.19}$$

when ϵ is assumed to follow $\mathcal{N}(0, \sigma_\epsilon^2)$.
In order to find the x^* solution of (IV.19), MCMC routines are used. This method is described in Gelfand and Smith (1990).

The MRM (Monotonous Reliability Method) Assume that $f : \mathbb{R}^d \to \mathbb{R}$ is a globally monotone, i. e. is monotone in each of its variables. Assume also that the set S of solutions of the equation $f(x) = 0$ is a continuous and simply (or one) connected set.
Assume for example that f is increasing on each of its variables. At each step, choose one point x in the unexplored subset of D. When $f(x) > 0$ then all points $y > x$ (meaning $y_i > x_i$ for all $1 \leq i \leq d$) are discarded from the unexplored region.
In the same way, when $f(x) < 0$, discard all the regions $\{y : y < x\}$.
Iteration of these steps produces an unexplored domain which shrinks to S.
Various ways of choosing x in the unexplored domain define specific algorithms. See Biret et al. (2015).

4.1.2 Outlook of the SAFIP algorithm

Basic features and properties We start with the iteration of the equivalence

$$(f(x) = 0) \iff \left(f(x) + \frac{x}{2k} + \frac{x}{2k} = \frac{x}{k}\right), \tag{IV.20}$$

which holds where $d = 1$, for any $k \neq 0$; for sake of convenience state $k > 0$.
We proceed defining a recurrence in the RHS in (IV.20), namely define a sequence $(z_i)_{i\in\mathbb{N}}$ with $z_i \in D$ and such that

$$z_{i+1} = z_i + \frac{z_{i-1} - z_i}{2} + kf(z_i). \tag{IV.21}$$

Defining

$$R_i = |z_i - z_{i-1}|, \tag{IV.22}$$

we obtain from (IV.21)

$$R_{i+1} \leq \frac{R_i}{2} + k|f(z_i)|. \tag{IV.23}$$

When $d > 1$, we may write

$$R_i = ||z_i - z_{i-1}||.$$

Thus, any sequence (z_i) which satisfies (IV.21) also satisfies (IV.23). We define $R_0 > 0$ arbitrary.
We now propose to substitute (IV.21) by a random sequence (z_i) which satisfies (IV.23). Also some additional conditions on (z_i) will be imposed. We will thus be able to prove the convergence of the resulting sequence (z_i) to some point in S ; reciprocally, for any x in S, when z_0 is close enough to x, the limit point of (z_i) will coincides with x.
Define z_0 and z_1 uniformly in D and $R_1 = ||z_1 - z_0||$.

For $i \geq 1$ compare $f(z_i)$ and $f(z_{i-1})$. Let $C \in [\frac{1}{2}, 1]$. If

$$|f(z_i)| \leq C|f(z_{i-1})|, \tag{IV.24}$$

then obtain z_{i+1} by

$$z_{i+1} := z_i + u_i, \tag{IV.25}$$

where u_i is randomly drawn on $\mathcal{B}\left(\underline{0}, \frac{R_i}{2} + k|f(z_i)|\right)$, where $\mathcal{B}(\omega, r)$ is the ball with center ω and radius r.
If (IV.24) is not fulfilled then the sequence $(z_j)_{j\in\mathbb{N}}$ stops. Draw then z_0 and z_1 again.
At this point we state

Theorem 1. *Any infinite sequence (z_i) defined as above converges a. s. with limit in S.*

We now add a number of conditions on the sequence (z_i) which entail that any point in S is reached asymptotically.
Let $x \in S$ and set $z_0 \in \mathcal{B}(x, \epsilon_0) = \{z : ||z - x|| \leq \epsilon_0\}$ for some $\epsilon_0 > 0$. Define further

$$E_0 := B \cap \{z : ||z - z_0|| > k_1|f(z_0)|\}, \tag{IV.26}$$

with $0 < k_1 < k$ and such that $k_1|f(z_0)| < 2\epsilon_0$; B is the ball with center z_0 and radius $\frac{R_0}{2} + k|f(z_0)|$. Therefore, E_0 is an annulus around z_0.
Let

$$A_1 = int\{\mathcal{B}(x, \epsilon_0) \cap B\}. \tag{IV.27}$$

By its very definition, the set A_1 is not void.
Assume that f satisfies the following regularity conditions

1. For all $x \in S$, there exists some $\epsilon_0(x) > 0$ such that if $z_0, z_1 \in \mathcal{B}(x, \epsilon_0)$ and $||x - z_1|| \leq ||x - z_0||$ then
$$\{z : |f(z)| \leq |f(z_1)|\} \subsetneq \{z : |f(z)| \leq |f(z_0)|\}.$$
2. There exists $0 < m < \frac{1}{4\epsilon_0}$ such that for all $x \in S$, for all $z_0 \in \mathcal{B}(x, \epsilon_0)$ for all $0 < \epsilon < k/2$, for all $z \in E_0 \cap A_1$,
$$|f(z_0)| - |f(z)| \geq m||z - z_0||.$$
By condition (1), the LHS in this inequality is non negative.

We then have

Theorem 2. *Let $x \in S$ and $\epsilon_0 > 0$ such that (1) and (2) hold. When $z_0 \in \mathcal{B}(x, \epsilon_0)$, the sequence (z_i) is infinite and satisfies Theorem 1. Furthermore* $\lim z_i = x$ *a. s.*

In order to cover all S by the limiting points of such sequences we also propose to add a step where we randomly select p points uniformly in D. These points are initial points of new sequences ; this allows to obtain a good covering of S by the limits of all these sequences.
Obviously this latest step does not substitute the entire algorithm ; clearly a hudge number of such points will approximate S from the start, the most inefficient Monte-Carlo random search method.

The stopping rule is defined through the definition of an accuracy index call *tol*. Define N the number of points to be reached in S. We may decide to stop the algorithm when N sequences (z_i) are such that the extremities are in S up to the accuracy, denoted *tol* in the sequel.

Enhanced algorithm In order to improve the coverage of S, keeping the same set of points z_0, we propose to modify the choice of z_{i+1} as given in (IV.24) and (IV.25) as follows. From $z_0, \ldots, z_i$ we build indeed i chains, each one starting from $z_j, 1 \leq j \leq i$. Obviously the sequence starting at z_i is as described above ; the new $i-1$ ones spread and develop in all directions. Any of these chains inherit of the properties mentioned in Theorem 1. Also, any x in S is asymptotically reached by one of those sequences, as i increases.
The sequences defined by an algorithm may be finite ; indeed condition (IV.24) may not hold for (z_{i-1}, z_i) and therefore z_{i+1} cannot be simulated. Thus no point z_{i+1} will be simulated since his father would be higher than his grandfather.
However his grandfather z_{i-1} is indeed lower than his grand-grandfather ; therefore his grandfather may have offspring. This grandfather is the root of a new generation, hence a new z_i which may satisfy (IV.24). In the same way all ancestors of z_{i-1} satisfy (IV.24) and are eligible for fatherhood.
We call a step of the algorithm the generation of all the offspring of the eligible points in the existing population of points. Such a step is followed by the generation of p uniformly distributed points in D as done in the basic algorithm.

In the sequel, we focus on the basic algorithm described in Section 4.1.2.

Reducing the computational cost tuning the parameters Firstly this algorithm makes use of very few parameters. Furthermore those can be tuned easily according to the complexity of the problem at hand. Indeed these parameters can be interpreted in connection with the computational burden. In some cases the function f is very costly and running an algorithm for a long time, without evaluating f often, may be of great advantage. Sometimes the function f is easy to calculate and the need is to get a quick description of S. Tuning k, C and m, together with the number of initiating points, makes use for those choices.
The following examples illustrate the role of each of the parameters, all the other ones being kept fixed.
The number of solutions which we require in the tolerance zone around S is fixed to 1000, but in the last example where the algorithm is evaluated with respect to this number.
Examples are presented in dimension 2. Higher dimension examples are presented in Section 4.1.2. Red points are couples (x_1, x_2) such that $f(x_1, x_2) > 0$. Points with negative values of f are blue. Black points are all blue or red ones whose f value belongs to $[-tol, tol]$.
Each example is summarized by three indicators. The first one is the runtime. The second one is the efficiency coefficient (EC) which is the ratio between the total number of evaluations of f and the number of solutions, which equals 1000 in all but the last example. This indicator is a measure of the number of calls to f which are required in order to obtain one solution to the equation $f(x) = 0$. The third indicator is of visual nature; in all those examples which are in dimension 2, the quality of the coverage of S can be considered qualitatively.

Remark 1. *The most important indicator is EC, since in all industrial applications, what really matters is the cost in evaluating f.*

The initialization step Call n the number of initiating points z_0, randomly selected on D. This is the initial cost of the method since the function f will be evaluated n times. Due to section 4.1.2, n should not be too large.

Example 1. *Let f be a bivariate function defined by*

$$(x_1, x_2) \mapsto f(x_1, x_2) = x_1^2 + x_2^2 - 0.5$$

The aim is to find $N = 1000$ pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy. All parameters but n are fixed. The tolerance is 0.01; the value of C is fixed being 0.75; the value of k is 1; the number p of supplementary points at each step of the algorithm is 1.
The solutions are close to $S = \{(x_1, x_2), f(x_1, x_2) = 0\}$, the circle with center $(0, 0)$ and radius

$\sqrt{0.5}$. *In Figure IV.19(a), the function f is intersected by the horizontal plane $z = 0$. The Figure IV.19(b) represents the intersection in the variables frame. The circle is then clearly visible. In Figures IV.20 (a), (b), (c), we have considered respectively $n = 5$, $n = 100$ and*

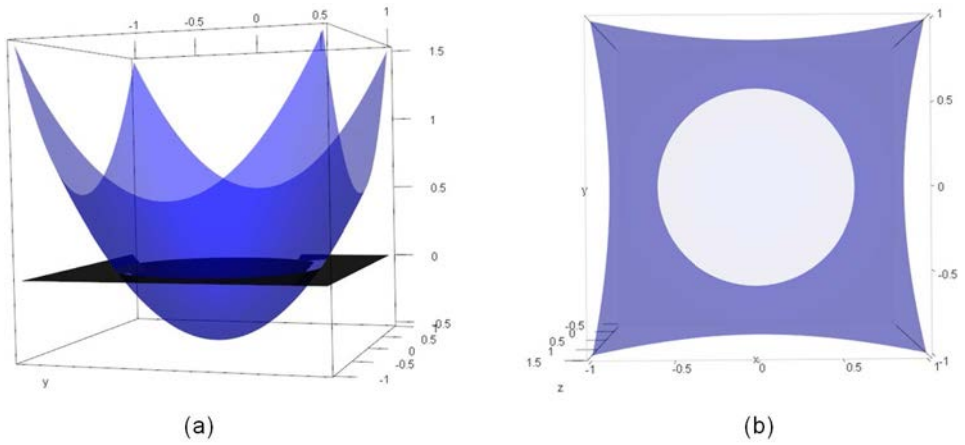

Figure IV.19 – Representations of the quadratic function

$n = 300$. *Clearly the more numerous the initial points, the more the number of chains, and*

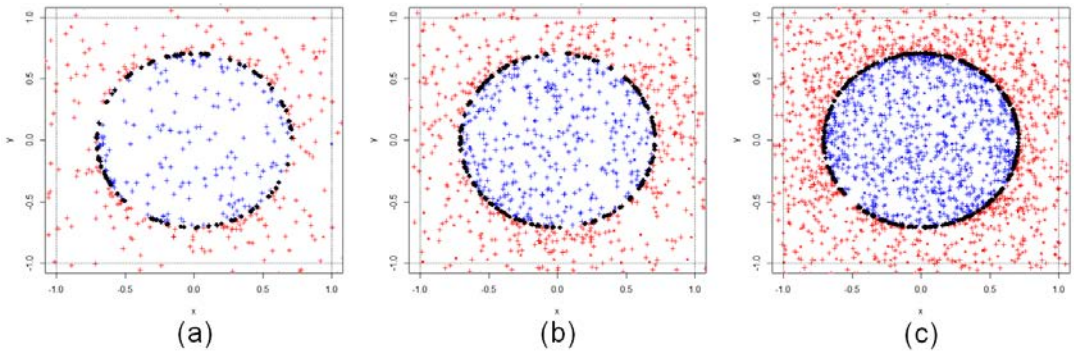

Figure IV.20 – Solving quadratic equation using SAFIP for three values of n

therefore the more numerous the points where the function f is evaluated; so the algorithm is costly as n increases. At the contrary, the better the coverage of S. Results are gathered in Table IV.1.

The rate of convergence The value of C pertains to the rate of convergence of the algorithm. Assume C small (C close to $1/2$) ; thus condition (IV.24) is rarely satisfied. The selected points will define chains with a fast convergence to S. However in order to satisfy (IV.24), many simulations in the ball B are required, leading to an increased runtime.

n	tol	N	C	k	p	Time	EC	Coverage
5	0.01	1000	0.75	1	1	0.32s	4.33	-
100	0.01	1000	0.75	1	1	0.60s	6.32	+
300	0.01	1000	0.75	1	1	1.54s	9.14	++

Tableau IV.1 – Results for Example 1 with different values of n

Example 2. *Let f be a bivariate function defined by*

$$(x_1, x_2) \mapsto f(x_1, x_2) = x_1^4 + x_2^3 - 0.5$$

The aim is to find $N = 1000$ pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy. All parameters but C are fixed. The number of initial points is 10; the tolerance is 0.015; the value of k is 1; the number p of supplementary points at each step of the algorithm is 1.
In Figure IV.21(a), the function f is intersected by the horizontal plane $z = 0$. The Figure IV.21(b) represents the intersection in the variables frame. In Figures IV.22 (a), (b), (c), we

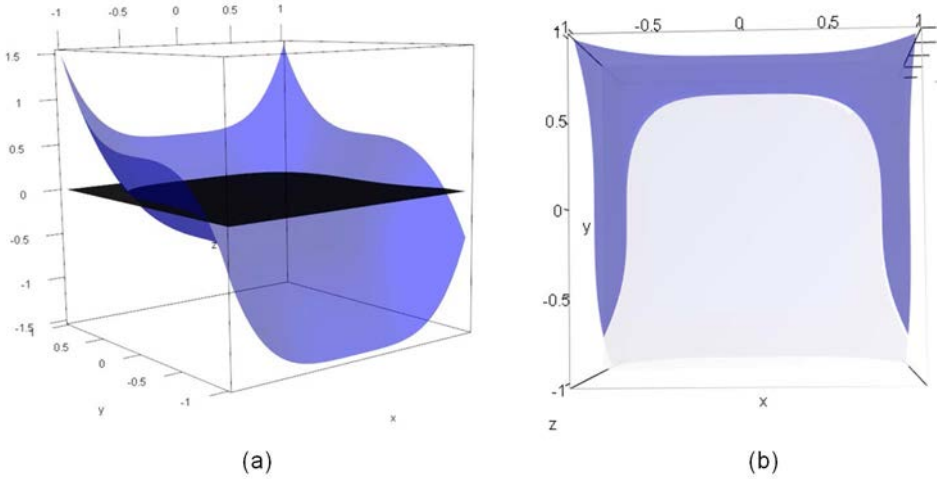

Figure IV.21 – Representations of the function with a chair shape

have considered respectively $C = 0.55$, $C = 0.75$ and $C = 0.95$. The greater C, the less the number of evaluations of f; furthermore the runtime decreases as C increases. Results are gathered in Table IV.2.

The role of k The parameter k is crucial for the simulation around z_i. In order to give some insight on the value of k, suppose that z belongs to $[-1, 1]^2$, and that the mean value of $|f(z)|$ is $\bar{f} = 10$. The current radius of the ball B is $\frac{R}{2} + k|f(z)|$, with R the distance between two

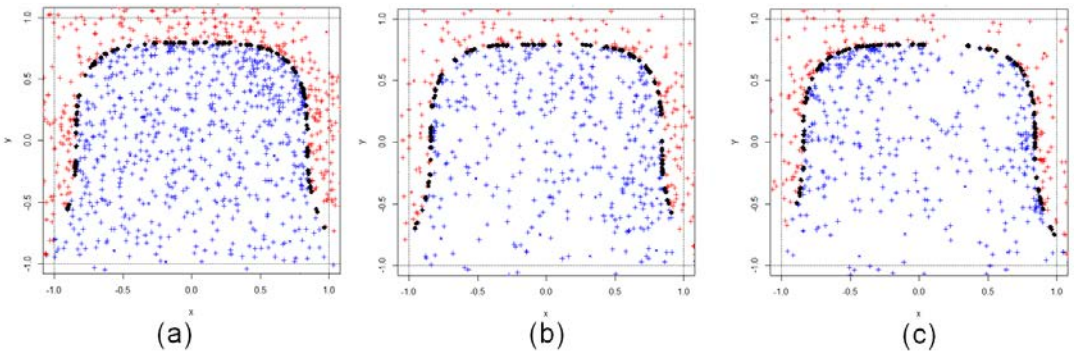

Figure IV.22 – Solving equation for the function with a chair shape using SAFIP for three values of C

n	tol	N	C	k	p	Time	EC	Coverage
10	0.015	1000	0.55	1	1	0.62s	8.36	+
10	0.015	1000	0.75	1	1	0.44s	5.33	+
10	0.015	1000	0.95	1	1	0.42s	5.05	+

Tableau IV.2 – Results for Example 2 with different values of C

points in the chain. Thus k should be at most of order $\frac{1}{f}$; in this way the ball B lays in $[-1,1]^2$, roughly.
This appears clearly in Example 3.

Example 3. *Let f be a bivariate function defined by*

$$(x_1, x_2) \mapsto f(x_1, x_2) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 - 50$$

The aim is to find $N = 1000$ pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy. All parameters but k are fixed. The number of initial points is 10 ; the tolerance is 3 ; the value of C is 0.75 ; the number p of supplementary points at each step of the algorithm is 1.
In Figure IV.23(a), the function f is intersected by the horizontal plane $z = 0$. Figure IV.23(b) represents the intersection in the variables frame. The mean value of f is 200 and its variations belong to $[-50, 350]$. In Figures IV.24 (a), (b), (c), we have considered respectively $k = 1/200$, $k = 10/200$ and $k = 50/200$. As k increases, the runtime also increases as does the number of evaluations of f in order to obtain one solution, and also the coverage of S improves. When f is costly, k should be chosen small. Results are gathered in Table IV.3.

The role of p The number of intermediate points is important since it allows to explore new points of D in quest for S. This number should be chosen small with respect to the number

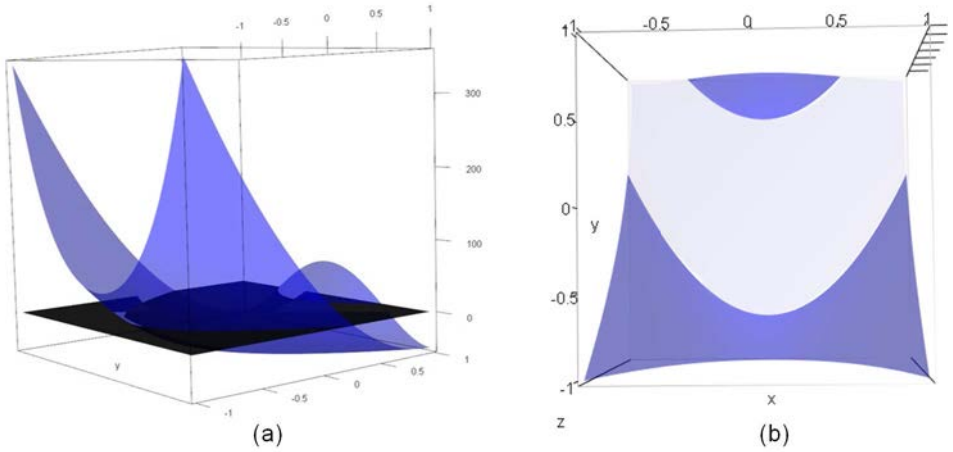

Figure IV.23 – Representations of the Rosenbrock function

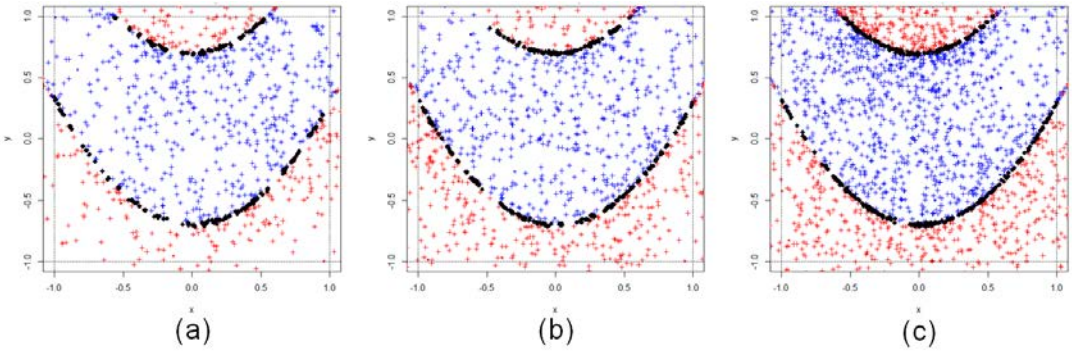

Figure IV.24 – Solving equation for the Rosenbrock function using SAFIP for three values of k

n of initializing points. The following example shows that very small values of p may be good choices.

Example 4. *Let f be a bivariate function defined by*

$$(x_1, x_2) \mapsto f(x_1, x_2) = (x_1 - 0.5)^2 + 3x_1x_2 - x_2^3 - 2.25$$

The aim is to find $N = 1000$ pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy.

n	tol	N	C	k	p	Time	EC	Coverage
10	3	1000	0.75	0.005	1	0.76s	10.69	+
10	3	1000	0.75	0.05	1	2.76s	18.71	+
10	3	1000	0.75	0.25	1	4.16s	48.49	++

Tableau IV.3 – Results for Example 3 with different values of k

All parameters but p are fixed. The number of initial points is 10 ; the tolerance is 0.04 ; the value of C is 0.75 ; the number k is 0.25.
In Figure IV.25(a), the function f is intersected by the horizontal plane z = 0. Figure IV.25(b) represents the intersection in the variables frame. p is chosen as 1, 3 and 5. In Figures IV.26

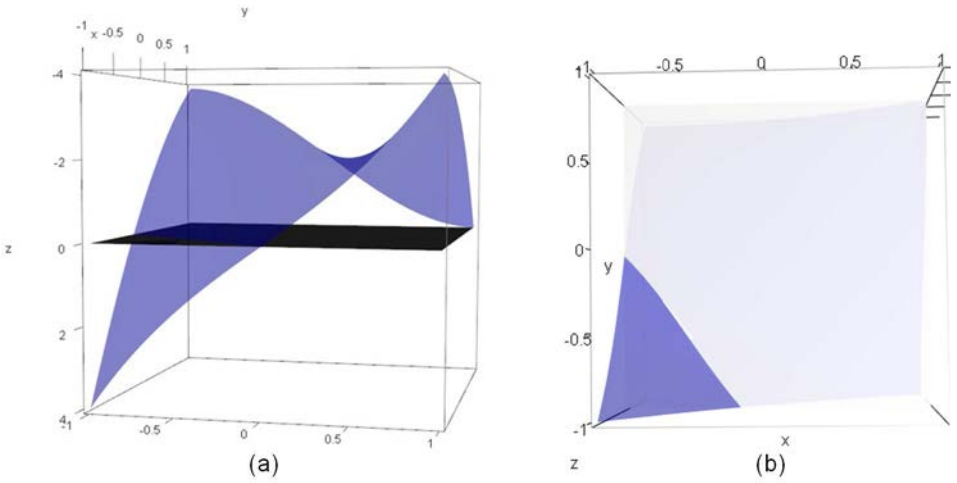

Figure IV.25 – Representations of the polynomial function

(a), (b), (c), we see that the algorithm has produced some insight to elements in S at the north-east region ; however, the 1000 solutions have been obtained on the south-west component of S. Having asked for more solutions, we would have obtained the north-east component. Increasing p to 3 or 5, the coefficient EC increases noticeably and the coverage of S clearly increases. Results are gathered in Table IV.4.

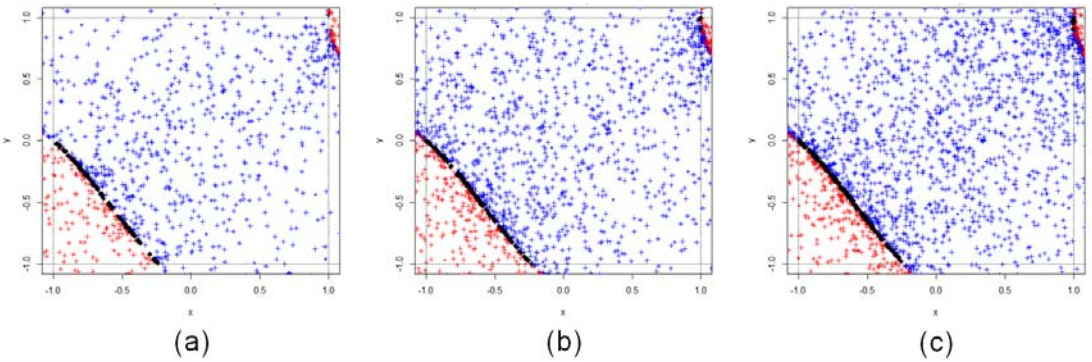

Figure IV.26 – Solving equation for the Rosenbrock function using SAFIP for three values of p

The tolerance factor tol The strongest the tolerance (i. e. when *tol* is small), the highest the number of evaluations of f, and the longest the runtime.

n	tol	N	C	k	p	Time	EC	Coverage
10	0.04	1000	0.75	0.25	1	2.12s	15.94	+
10	0.04	1000	0.75	0.25	3	3.24s	14.58	+
10	0.04	1000	0.75	0.25	5	4.96s	17.01	++

Tableau IV.4 – Results for Example 4 with different values of p

Example 5. *Let f be a bivariate function defined by*

$$\begin{aligned}(x_1, x_2) \mapsto f(x_1, x_2) = {} & 8\sin(7(x_1 - 0.9)^2)^2) + 6\sin((14(x_1 - 0.9)^2)^2) + (x_1 - 0.9)^2 \\ & + 8\sin((7(x_2 - 0.9)^2)^2) + 6\sin((14(x_2 - 0.9)^2)^2) \\ & + (x_2 - 0.9)^2 - 15\end{aligned}$$

The aim is to find $N = 1000$ pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy. All parameters but tol are fixed. The number of initial points is 10 ; the value of C is 0.75 ; the number k is 0.08 ; the number p of supplementary points at each step of the algorithm is 1.
In Figure IV.27(a), the function f is intersected by the horizontal plane $z = 0$. Figure IV.27(b) represents the intersection in the variables frame. The function oscillates between -15 and 15.

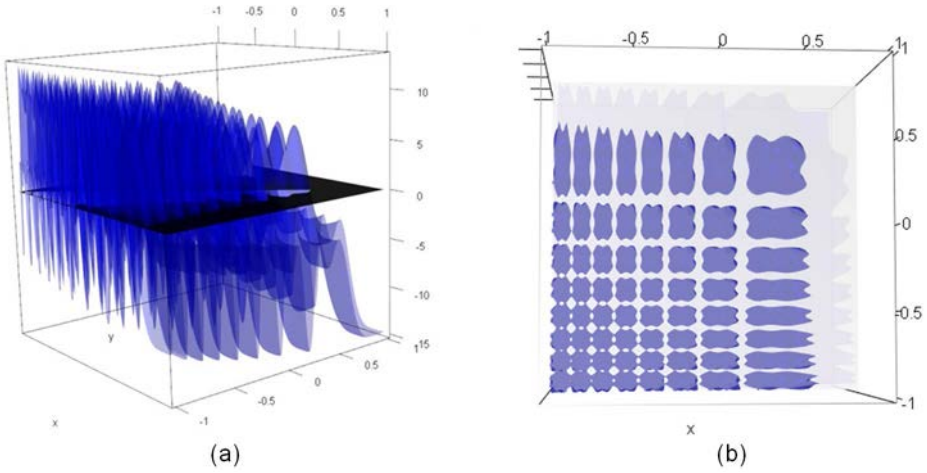

Figure IV.27 – Representations of the trigonometric function

In Figures IV.28 (a), (b), (c), algorithm results are illustrated for three values of tol : 0.15, 0.75 and 1.5 . When tol varies from 0.15 to 1.5, the coefficient EC gets divided by 2. Results are gathered in Table IV.5.

Due to the complexity of the function and of the set S, coverage is mild whatever *tol* ; it depends upon the required number of solutions only.

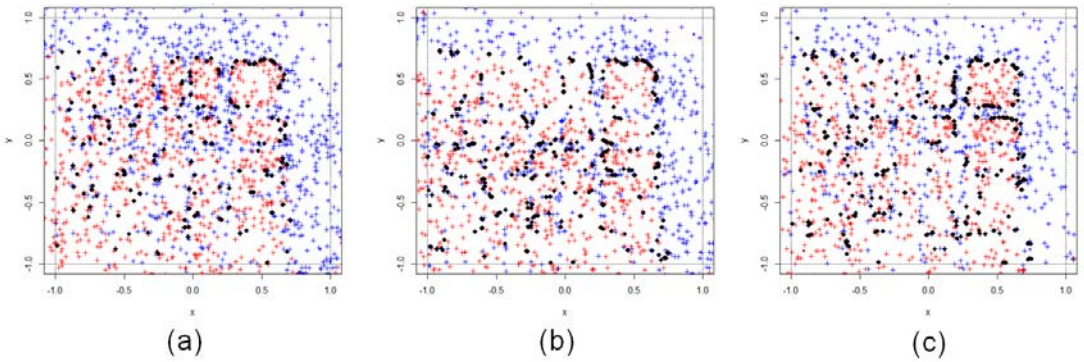

Figure IV.28 – Solving equation for the trigonometric function using SAFIP for three values of *tol*

n	tol	N	C	k	p	Time	EC	Coverage
10	0.15	1000	0.75	0.25	1	2.6s	43.47	-
10	0.75	1000	0.75	0.25	1	1.68s	32.2	-
10	1.5	1000	0.75	0.25	1	1.3s	22.85	-

Tableau IV.5 – Results for Example 5 with different values of p

The role of N, the required number of solutions The same function as in Example 4 is used in order to focus on the role of the number of solutions. When we ask for 15000 points in S, then the runtime remains quite satisfactory; the EC coefficient is 76, due to a choice of $n = 1000$. The coverage of S is quite fair. Clearly the quality of the solutions improves with

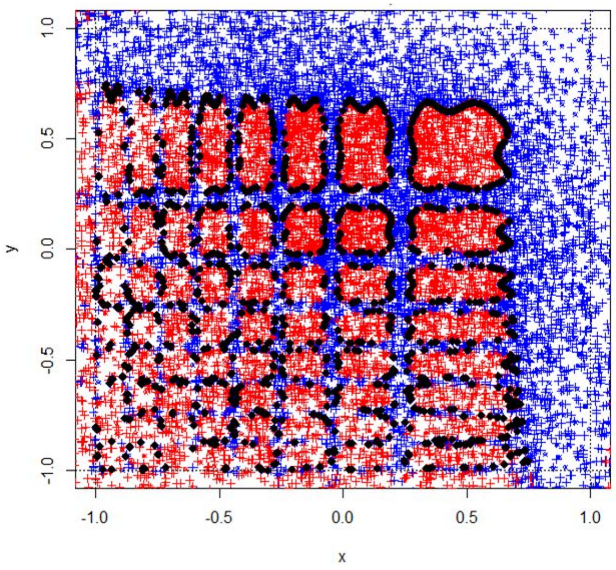

Figure IV.29 – Solving equation for the trigonometric function using SAFIP for a bigger number of required final points and a tolerance of 0.15

n	tol	N	C	k	p	Time	EC	Coverage
10	0.4	100	0.75	0.025	1	0.48s	55.33	-
10	0.4	1000	0.75	0.025	1	3.96s	60.64	-
10	0.4	2000	0.75	0.025	1	8.6s	83.82	-

Tableau IV.6 – Results for Example 6 with different values of N

the required number of solutions. Not only do we get more solutions, but the coverage of S improves noticeably.

Example 6. *Let f be a bivariate function defined by*

$$(x_1, x_2) \mapsto f(x_1, x_2) = 20 + x_1^2 - 10\cos(2\pi x_1) + x_2^2 - 10\cos(2\pi x_2) - 60$$

The aim is to find N pairs (x_1, x_2) such that $|f(x_1, x_2)| \leq tol$ where tol is the accuracy. All parameters but N are fixed. The number of initial points is 10 ; tol is fixed to 0.4 ; the value of C is 0.75 ; the number k is 0.025 ; the number p of supplementary points at each step of the algorithm is 1.
In Figure IV.30(a), the function f is intersected by the horizontal plane $z = 0$. Figure IV.30(b) represents the intersection in the variables frame. In Figures IV.31 (a), (b), (c), algorithm

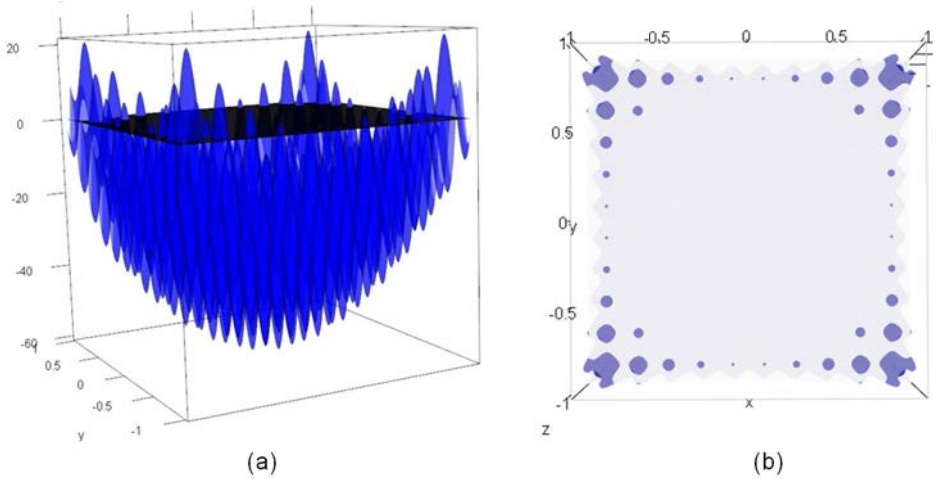

Figure IV.30 – Representations of the Rastrigin function

results are illustrated for three values of N : 100, 1000 and 2000. When N is small, the important feature of the result is that S is covered equally. So no cluster of solutions seems to appear ; this is important for exploratory analysis. Results are gathered in Table IV.6.

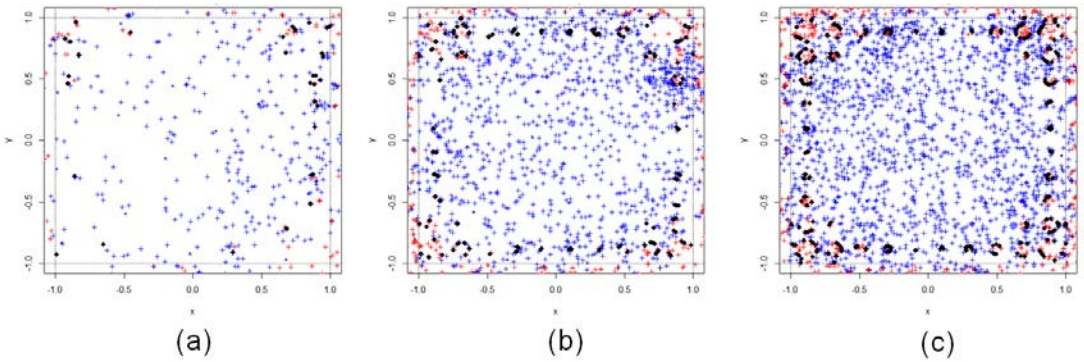

Figure IV.31 – Solving equation for the Rastrigin function using SAFIP for three values of N

Increasing the dimension We consider a collection of functions which mimick Example 1, increasing the dimension. The required number of solutions is kept as $N = 500$ in all cases. We firstly consider the case in dimension 3, namely we look at points situated in

$$S := \{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 - 0.5 = 0\}, \tag{IV.28}$$

with $-1 \leq x_i \leq 1$ for $i = 1, 2, 3$. The result appears in Figure IV.32. We also have considered

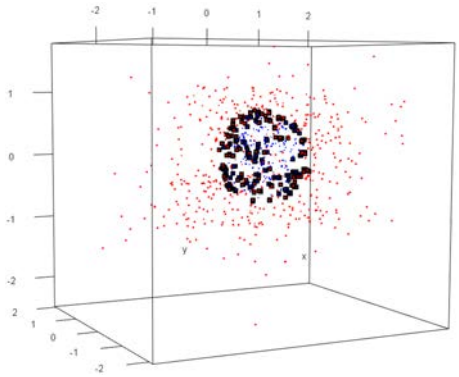

Figure IV.32 – Results for spheres in dimension 3

the set

$$S := \{(x_1, x_2, x_3) : max(x_1, x_2, x_3) - 0.5 = 0\}; \tag{IV.29}$$

See Figure IV.33.

Looking at similar examples as (IV.28), we consider $d = 4$ and $d = 10$; the results comparing three dimensions are in Table IV.7. The same is available for (IV.29) in Table IV.9. The number of initializing points has been chosen accordingly : $n = 75$ for $d = 4$, and $n = 1000$ for $d = 10$; a coherent choice for n would have been $n = 5^9$ for $d = 10$, an impracticable choice.

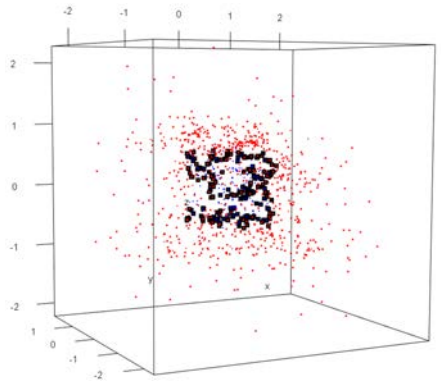

Figure IV.33 – Results for cubes in dimension 3

Dim	n	tol	N	C	k	p	Time	EC
2	5	0.1	500	0.75	1	1	0.22s	4.81
3	25	0.1	500	0.75	1	1	4.72s	6.64
4	75	0.1	500	0.75	1	1	0.4s	9.7
10	1000	0.1	500	0.75	1	1	53s	449

Tableau IV.7 – Results for spheres in different dimensions

Obviously the indicator EC increases with n. However, choosing $n = 5^9$ and $N = 500$, the value of EC exceeds 2000, which proves that n should be kept low, growing slowly with respect to d.

4.1.3 Simultaneous inverse problems

Algorithm Let f and g denote two functions defined on D; each of these functions f and g is assumes to satisfy hypothesis (IV.24) together with conditions (1) and (2). We will make use of constants C, k, n and p defined in Section 4.1.2; these constants will play a similar role in the present on f and g. The number of common solutions to the system

$$\begin{cases} f(x) &= 0 \\ g(x) &= 0 \end{cases} \tag{IV.30}$$

Dim	n	tol	N	C	k	p	Time	EC
2	5	0.1	500	0.75	1	1	0.16s	4
3	25	0.1	500	0.75	1	1	4.2s	5.04
4	75	0.1	500	0.75	1	1	0.72s	8
10	1000	0.1	500	0.75	1	1	51s	614

Tableau IV.8 – Results for cubes in different dimensions

is denoted N.
Also the present section considers simultaneous inverse problems pertaining to two functions ; quantization to a given number of functions is straightforward.
The algorithm is as follows with similar notation as in Section 4.1.2, it holds

$$\left\{\begin{array}{rcl} f(x) & = & 0 \\ g(x) & = & 0 \end{array}\right. \Leftrightarrow \left\{\begin{array}{rcl} f(x)+\frac{x}{2k}+\frac{x}{2k} & = & \frac{x}{2} \\ g(x)+\frac{x}{2k}+\frac{x}{2k} & = & \frac{x}{2} \end{array}\right. \tag{IV.31}$$

which yields to define

$$z_{i+1} = z_i + \frac{z_{i-1}-z_i}{2} + k\max(|f(z_i)|,|g(z_i)|). \tag{IV.32}$$

Inequality (IV.23) is substituted by

$$R_{i+1} \leq \frac{R_i}{2} + kmax(|f(z_i)|,|g(z_i)|). \tag{IV.33}$$

Similarly as in (IV.25), the choice of z_{i+1} follows the rule

$$z_{i+1} = z_i + u_i \tag{IV.34}$$

where u_i is drawn randomly on $\mathcal{B}(\underline{0}, \frac{R_i}{2} + kmax(|f(z_i)|,|g(x_i)|)$.
With those changes, denoting $S = \{x : f(x) = 0, g(x) = 0\}$, it holds

Theorem 3. *Any sequence (z_i) defined as above converges a. s. with limit in S.*

and

Theorem 4. *For any $x \in S$ and $\epsilon_0 > 0$ such that (1) and (2) hold simultaneously for f and g, and when $z_0 \in \mathcal{B}(x, \epsilon_0)$, thus the sequence (z_n) is infinite and converges to x.*

Examples Due to (IV.34), the point z_{i+1} is randomly chosen in a ball B centerd at z_i when both $|f(z_i)|$ and $|g(z_i)|$ share a common measural order of magnitude. The best case is when B has a moderate radius ; it is therefore useful to normalize f and g on D ; this preliminary procedure obviously does not modify the set S.
We present three examples of simultaneous inversion, based on the functions presented on Section 4.1.2. In all examples the parameters are $n = 20$, $p = 1$, $tol = 0.01$, $C = 0.75$, $k = 1$. N equals 10 in Example 7, it equals 100 in Example 8 and Example 9.

Example 7 (A regular case)**.** *We choose f as in Example IV.20 and $g(x) = f(x-a), a = (0.2, -0.2)$. Therefore $f(x) = 0$ is as in Example IV.20 and $g(x) = 0$ is a circle with same radius and center a.*

Figures IV.34(a) and (b) show the graphs of f and g together with the intersection of the plane $z = 0$. The set S consists in the two points shown in Figure IV.34(b). Those points are indeed

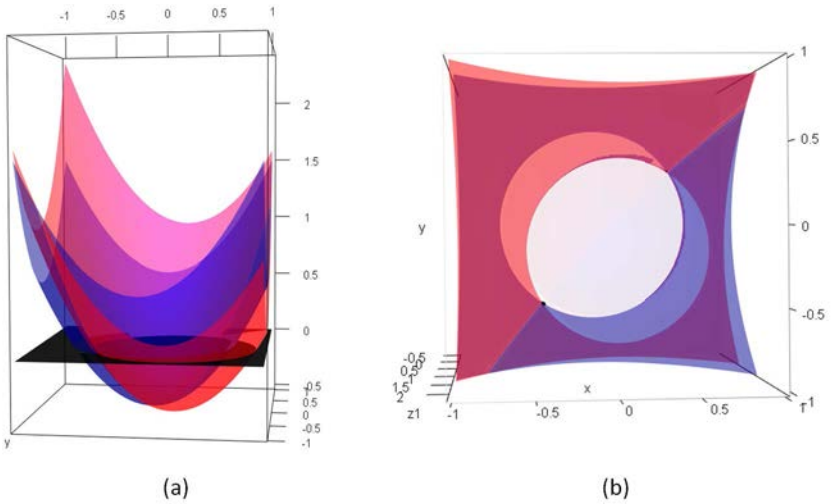

Figure IV.34 – Representations of f, g and S

well estimated by the present algorithm, as seen in Figure IV.35. The runtime is 0.62s and the

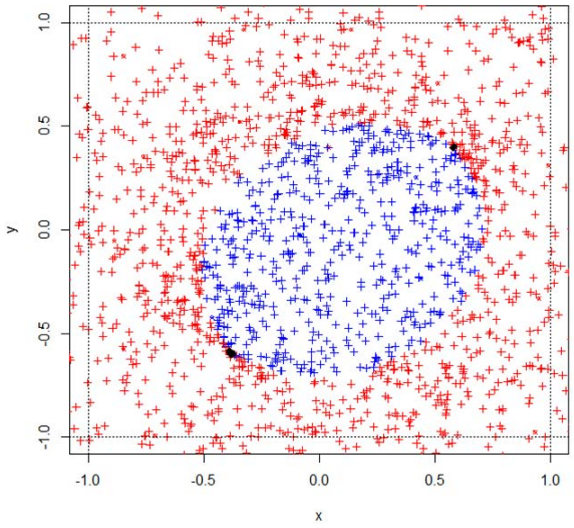

Figure IV.35 – Solutions obtained with SAFIP algorithm

efficiency coefficient is 516.

Example 8 (Mixing a regular function and an irregular one)**.** *We choose $f(x)$ as defined in Example IV.24, a regular function, and $g(x)$ the Rastrigin function of Example IV.31. The Figure IV.36(a) shows the two function, and Figure IV.36(b) provides the set S, which is defined as the intersection of the frontier points of the red domains (the solutions to $g(x) = 0$) wt=ith the frontier points of the blue domains (the solutions to $g(x) = 0$). There are 29 points in S. The algorithm provides solutions as shown in Figure IV.37, with runtime 14s and efficiency*

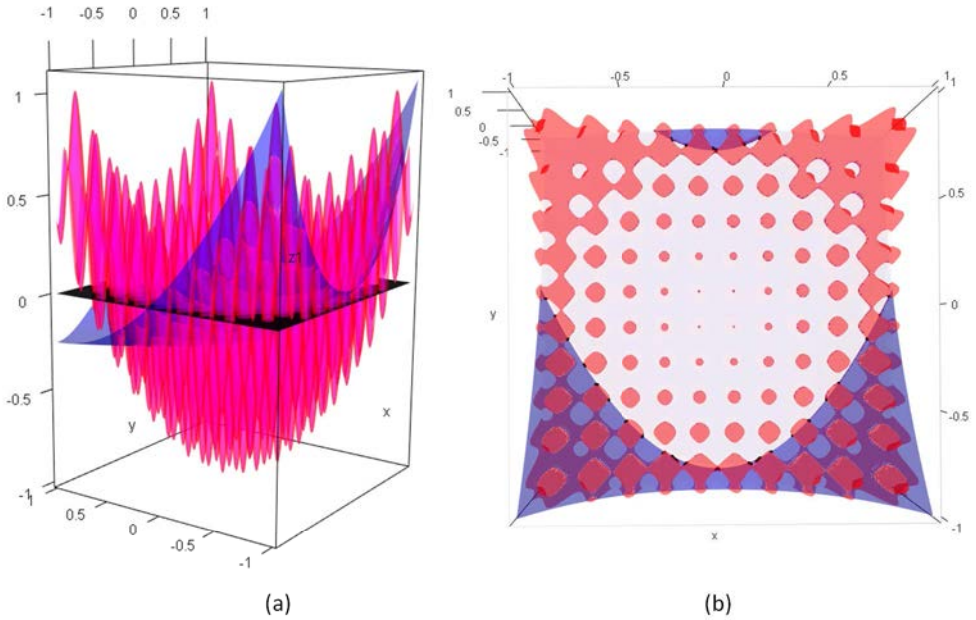

Figure IV.36 – Representations of f, g and S

coefficient 375. Table **??** *provides results for different values of C, k and n.*
As C increases, EC decreases; as k or n increases, EC increases too.
A clear feature in Figure IV.37 is that all the 29 points in S are obtained a limiting points of SAFIP.

Example 9 (A last example)**.** *We choose $f(x)$ as in Example IV.20 and $g(x)$ the trigonometric function of Example IV.28. Figure IV.38(a) shows the functions f and g; Figure IV.38 (b) shows the intersection set S which contains 33 points. We asked for $N = 100$ solutions; the set S is not totally covered (we obtain 26 points in S as it can be seen on Figure IV.39); a larger value of N would provide all solutions The runtime is 4.1s and EC is 1102.*

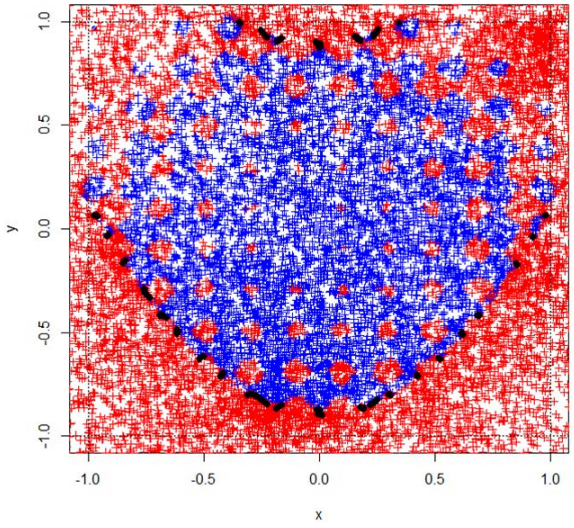

Figure IV.37 – Solutions obtained with SAFIP algorithm

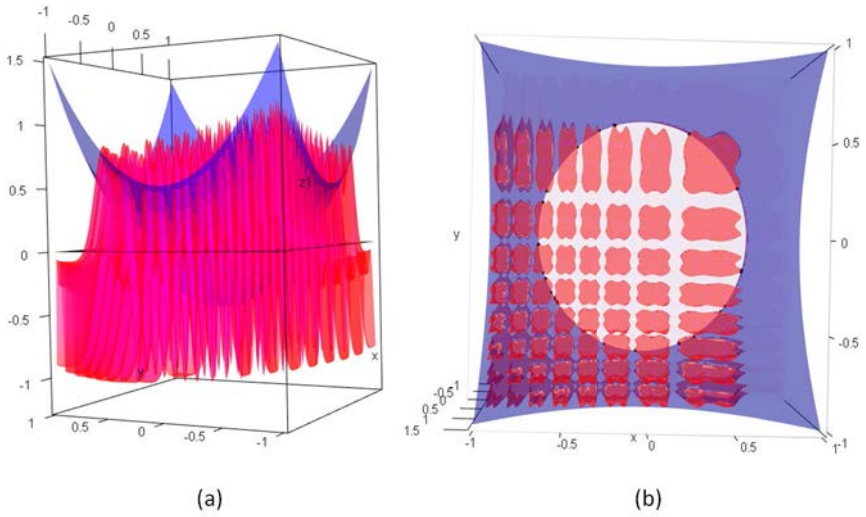

Figure IV.38 – Representations of f, g and S

4.1.4 Appendix

Proof of Theorem 1. Step 1. We prove that the sequence $(R_i)_{i\in\mathbb{N}}$ converges to 0 a. s. Denote $a := |f(z_0)| > 0$. By (IV.24),

$$|f(z_i)| \leq aC^i,$$

C	EC	Temps
0.55	905	4.72s
0.75	469	1.66s
0.95	311	1.24s
k	**EC**	**Temps**
1	546	5.02s
10	1963	8.8s
50	6372	32.04s
n	**EC**	**Temps**
10	577	2.54s
100	622	3.36s
300	708	3.36s

Tableau IV.9 – Results for cubes in different dimensions

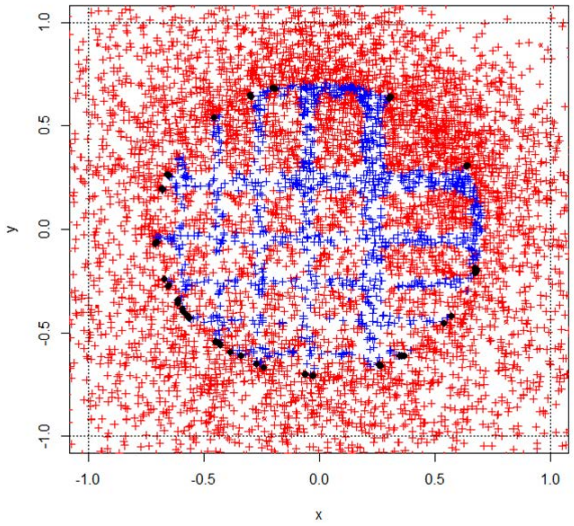

Figure IV.39 – Solutions obtained with SAFIP algorithm

hence $R_{i+1} \leq \frac{R_i}{2} + akC^i$.
The sequence $(R_i)_{i\in\mathbb{N}}$ is now compared to the sequence $(x_i)_{i\in\mathbb{N}}$ defined by

$$x_{i+1} = \frac{x_i}{2} + akC^i.$$

It holds

$$\begin{aligned} x_n &= \frac{x_0}{2^n} + \frac{ak}{2^{n-1}} + \frac{akC}{2^{n-2}} + \frac{akC^2}{2^{n-3}} + \ldots + \frac{akC^{n-2}}{2^1} + akC^{n-1} \\ &= \frac{x_0}{2^n} + akC^{n-1} \sum_{j=0}^{n-1} \left(\frac{1}{2C}\right)^j. \end{aligned} \tag{IV.35}$$

When $C > 1/2$, it follows that x_n given in (IV.35) tends to 0 as $n \to \infty$.
Since the generic term of $(R_n)_{n\in\mathbb{N}}$ satisfies

$$R_n \leq \frac{R_0}{2^n} + akC^{n-1} \sum_{j=0}^{n-1} \left(\frac{1}{2C}\right)^j, \tag{IV.36}$$

where the RHS is x_n, it follows that R_n tends to 0 as $n \to \infty$.

Step 2. Since $z_0 \in D$, a bounded set in $\mathbb{R}^d$, for any $i \in \mathbb{N}$ there exists a finite $M > 0$ such that $z_i \in \mathcal{B}(z_0, M)$. Hence $(z_n)_{n\in\mathbb{N}}$ is a bounded sequence.
By Bolzano-Weierstrass theorem, there exists some convergent subsequence $(z_{(i)})_{i\geq 0}$.
Assume at present that $(z_n)_{n\in\mathbb{N}}$ is an a. s. convergent sequence, and denote l its limit. We prove that l belongs to S. Indeed by (IV.21), writing $u_i = v_i(\frac{R_i}{2} + k\,|f(z_i)|)$ for v_i uniformly distributed on $\mathcal{B}(0,1)$, the unit ball in $\mathbb{R}^d$. Going to the limit in (IV.21), $l = l + \lim\limits_{i\to\infty} u_i$. It follows that $\lim\limits_{i\to\infty} \frac{R_i}{2} + k|f(z_i)| = 0$. Since $\lim\limits_{i\to\infty} R_i = 0$, it holds

$$\lim_{i\to\infty} |f(z_i)| = 0 \text{ a. s.}$$

By continuity of f, it follows that $\lim\limits_{i\to\infty} |f(z_i)| = f(l)$ and then $f(l) = 0$. We have proved that $l \in S$.
It follows that, if $(z_n)_{n\in\mathbb{N}}$ converges, then any converging subsequence $(z_{(i)})$ also converges to some point in S.
It remains to prove that $(z_n)_{n\in\mathbb{N}}$ converges, showing that it is a Cauchy sequence.
Let $(m, n) \in \mathbb{N}^2, m > n$. Then

$$\begin{aligned} \sup_{m>n} ||z_m - z_n|| &\leq \sup_{m>n} \sum_{j=n+1}^{m} ||z_j - z_{j-1}|| \\ &\leq \sup_{m>n} \sum_{j=n+1}^{m} r_j. \end{aligned}$$

By (IV.36),

$$\sup_{m>n}||z_m - z_n|| \leq \sup_{m>n}\sum_{j=n+1}^{m}\left(\frac{r_0}{2^j} + akC^{j-1}\left(\frac{2C - \left(\frac{1}{2C}\right)^{j-1}}{2C-1}\right)\right)$$
$$\leq \sup_{m>n}\left(\frac{r_0\left(1-\left(\frac{1}{2}\right)^{m-n}\right)}{2^n} + \frac{2akC^{n+1}}{2C-1}\times\frac{1-C^{m-n}}{1-C} - \frac{ak}{(2C-1)2^n}\times\frac{1-\left(\frac{1}{2}\right)^{m-n}}{\frac{1}{2}}\right),$$

with $0 < 2C-1 < 1$. Since $m > n$ and $C < 1$

$$\sup_{m>n}||z_m - z_n|| \leq \frac{r_0}{2^{n+1}} + \frac{2akC^{n+1}}{(2C-1)(1-C)} - \frac{ak}{(2C-1)2^{n-1}}$$

and therefore

$$\lim_{n\to\infty}\sup_{m>n}||z_m - z_n|| = 0, \tag{IV.37}$$

which proves the claim.

■

Proof of Theorem 2. By (IV.26), we have $E_0 = \{z : k_1|f(z_0)| \leq ||z - z_0|| \leq \frac{R_0}{2} + k|f(z_0)|\{$, with $\epsilon_0 = ||x - z_0||$. We have to prove that $E_0 \cap A_1 \neq \emptyset$.
By (IV.27) and since $E_0 \subset B$, this is equivalent to prove that $\mathcal{B}(x,\epsilon_0)\cap E_0 \neq \emptyset$. By the definition of E_0 which is an annulus centred on z_0 with a minimal radius of $2\epsilon_0$ and since $z_0 \in \partial\mathcal{B}(x,\epsilon_0)$ according to the definition of ϵ_0, $\mathcal{B}(x,\epsilon_0)\cap E_0 \neq \emptyset$ and so $E_0 \cap E_1 \neq \emptyset$.
Let $z_1 \in A_1 \cap E_0$. we prove that z_1 satisfies (IV.24).
By condition 2, it follows

$$|f(z_0)| - |f(z_1)| \geq mk_1|f(z_0)|,$$

since $z_1 \in E_0$. This is equivalent to

$$|f(z_1)| \leq (1 - mk_1)|f(z_0)|$$

With an arbitrary k_1 close to 0 such that $0 < mk_1 < \frac{1}{2}$. Getting $C = (1 - mk_1) \in [\frac{1}{2}, 1]$, we have $|f(z_1)| \leq C|f(z_0)|$ for $z_1 \in E_0$. Thus $z_1 \in A_1 \cap \{z_1, |f(z_1)| \leq C|f(z_0)|\}$ and z_0 can have an offspring.
Iterating the above argument we can construct a sequence of balls $\mathcal{B}(x,\epsilon_i)$ with lower bounded and decreasing sequence of radius. Thus this sequence converges to some limit. By Theorem 1, $\lim_{i\to\infty} zi = x^* \in S$.

We show that $x^* = x$ by contradiction.
If $x^* \neq x$, thus there exists $i \in \mathbb{N}$ such that $x \notin \mathcal{B}(x^*, ||x^* - z_i||)$. But z is simulated around x with decreasing radius to 0. Hence is the contradiction. Thus $x^* = x$ and we have proved Theorem 2. ■

4.2 Modification de la méthode pour réduire le nombre d'appels à la fonction : méthode COMET

On note $\hat{f}$ l'approximation de f par un estimateur à noyau, méthode de méta-modélisation décrite au Chapitre II. La méthode COMET est basée sur un algorithme général (cf. Algorithme 2) que les méthodes déterministes par grilles et la méthode probabiliste par modèle.

Choix des paramètres
n : nombre de points évalués initialement.
k : nombre de candidats testés à chaque itération.
K : noyau (gaussien par exemple).
n_t : nombre de points dans la liste tabou.
tol : tolérance.
n_f : nombre de points finaux dans la zone de tolérance $[-tol, tol]$.
Initialisation
Simuler n points $x_1, \ldots, x_n$ suivant une loi uniforme sur le domaine d'entrée D et évaluer $f(x_1), \ldots, f(x_n)$.
$i \leftarrow 0$.
Tant que (le critère d'arrêt n'est pas atteint) **faire**
  $i \leftarrow i + 1$.
  Simuler k points $z_1, \ldots, z_k$ suivant une loi uniforme sur D.
  Pour j **de** 1 **à** k **faire**
    Calculer $\hat{f}(z_j)$.
  Fin Pour
  Choisir $z_{m_1}, \ldots, z_{m_{n_t}}$ tel que $|\hat{f}(z_{m_1})| = \min_{1 \leq j \leq k} |\hat{f}(z_j)| \leq |\hat{f}(z_{m_2})| \leq \ldots \leq |\hat{f}(z_{m_{n_t}})| \leq |\hat{f}(z_j)|$, pour $j \notin \{m_1, \ldots, m_{n_t}\}$.
  Choisir le z_m parmi $z_{m_1}, \ldots, z_{m_{n_t}}$ assurant la meilleure répartition avec les x_i selon un critère maximin $\max_{1 \leq l \leq n_t} \min_{1 \leq j \leq N_0} ||x_j - z_{m_l}||$.
  $x_{n+i} \leftarrow z_m$.
  Évaluer $f(x_{n+i})$.
Fait
Conclusion
Renvoyer tous les points qui se trouvent dans la zone de tolérance.

Algorithme 3 – Algorithme initial de la méthode COMET

Les particularités de la méthode COMET sont décrites dans l'algorithme représenté à la Figure IV.40. Le critère d'arrêt peut être basé sur le fait que l'algorithme n'évolue plus (les

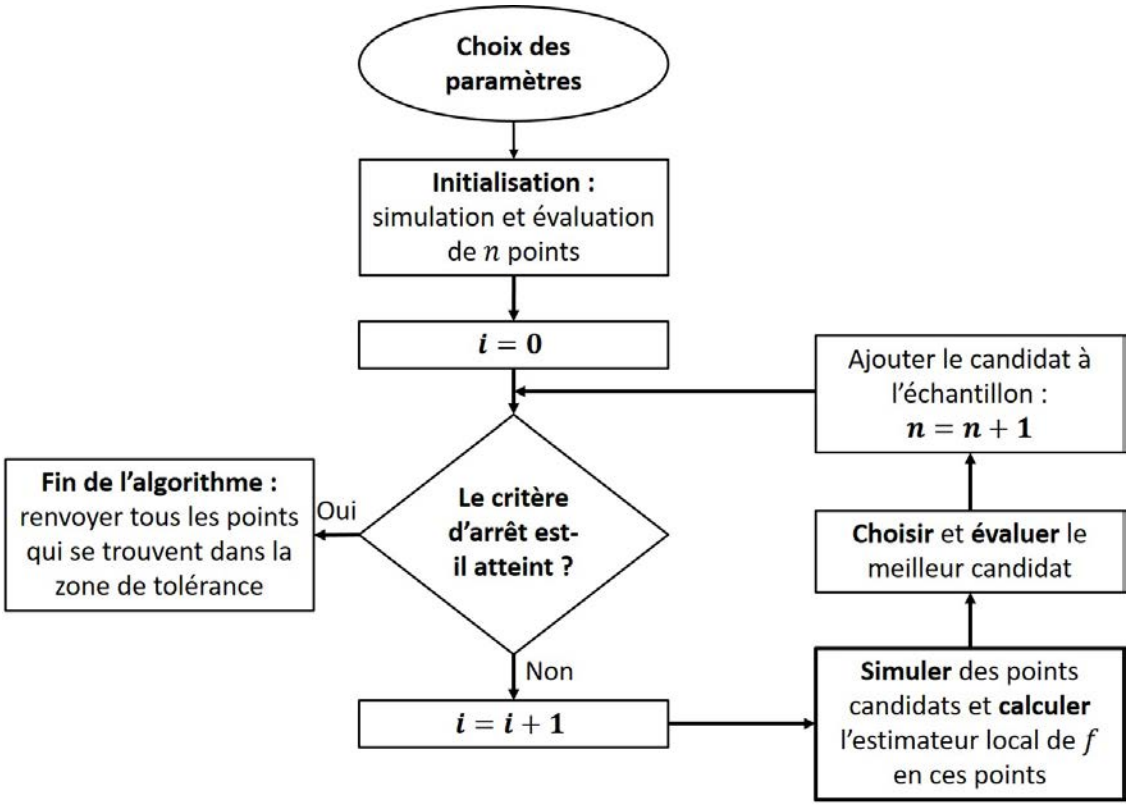

Figure IV.40 – Algorithme initial de la méthode COMET

nouveaux points sont toujours dans la zone de tolérance ou n'améliorent pas la valeur de f) ou sur le fait d'atteindre le nombre de points finaux dans la zone de tolérance (nombre de points et tolérance choisis par l'utilisateur).
Comme on cherche des points dans un ensemble infini, le premier choix de critère d'arrêt est difficile à atteindre. Le second est très utile en pratique puisque l'utilisateur peut récupérer ce dont il a besoin, pas plus, pas moins.

Cette méthode peut être utilisée en toute dimension et même directement sur un code de calcul puisqu'aucun pré-traitement sur la fonction n'est exigé. Cependant, la convergence est difficile pour des petites tolérances. En effet, par le fait que la méthode est stochastique, la probabilité de tirer des points dans une zone très réduite est faible. De plus, l'efficacité de la méthode (nombre d'appels à la fonction pour obtenir un point solution) dépend fortement de la dimension, de la tolérance mais aussi de la régularité de la fonction et de celle de la solution

(taille, continuité, nombre d'ensembles connexes). Nous représentons cette efficacité par un indice noté CE et correspondant au nombre d'appels à la fonction pour obtenir un point dans la zone de tolérance choisie. Plus l'indice est grand, moins la méthode est efficace.

4.3 Test de la méthode modifiée sur des fonctions usuelles

On teste cette méthode sur des fonctions usuelles (similaires à celles de la section précédente). Ceci permet de montrer les nombreuses possibilités de la méthode mais aussi les variations importantes des temps de calculs et de l'indice CE.

Dans toutes les fonctions proposées, nous travaillons en dimension 2 dans le domaine $[-1,1]^2$. Comme certaines solutions peuvent être très proches des bords du domaine, nous l'élargissons légèrement afin de pouvoir les atteindre. Les algorithmes fonctionneront donc sur $[-1.1,1.1]^2$ mais les solutions conservées seront bien dans $[-1,1]^2$.
Pour chacun des exemples suivants, nous utilisons les réglages suivants :

- 100 points initiaux,
- 100 points candidats à chaque itération,
- 10 points dans la liste tabou.

Fonction quadratique En dimension 2, la fonction quadratique est de la forme suivante :

$$f(x,y) = x^2 + y^2$$

On veut obtenir des points proches de la solution $S = \{(x,y) : f(x,y) = 0.5\}$, ce qui correspond à la situation de la Figure IV.41. A gauche, on a la représentation de f en bleu coupée par le plan horizontal $z = 0.5$ en noir. A droite, on a la représentation de l'intersection dans le plan (x,y). Les solutions S forment le cercle de centre $(0,0)$ et de rayon $\sqrt{0.5}$ que l'on voit à droite sur la Figure IV.41. L'ordre de grandeur de $f(x,y)$ est 1 (moyenne entre les valeurs absolues du min et du max). On cherche à obtenir 500 points dans une zone de tolérance égale à 1% de l'ordre de grandeur.
On obtient les points solutions en noir sur le premier graphique de la Figure IV.42, en rouge sur le second. Les points rouge et bleu sur la figure de gauche sont les points évalués lors de l'algorithme. La méthode permet d'obtenir un grand nombre de points proches de la solution S qui représentent bien S, comme on peut le voir sur la figure de droite avec la représentation du cercle.

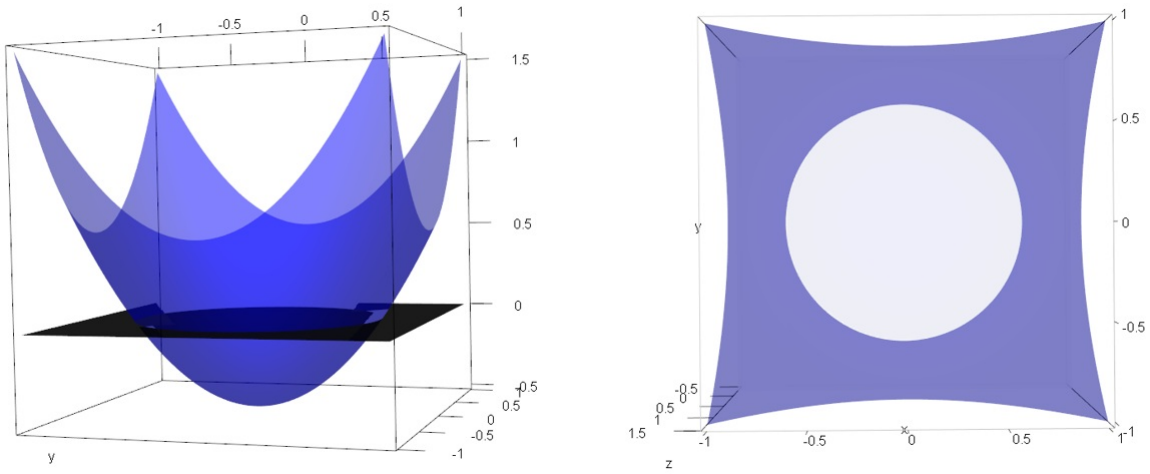

Figure IV.41 – Représentation de la fonction quadratique intersectée au niveau 0.5

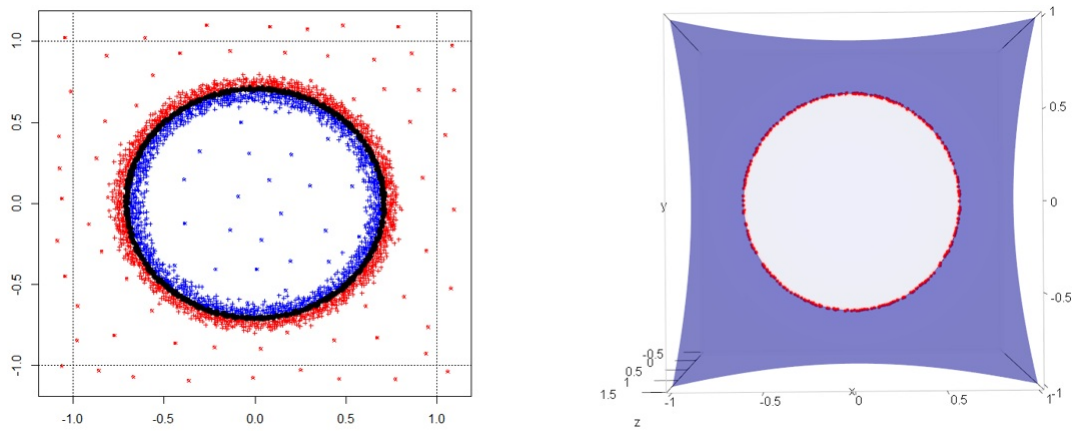

Figure IV.42 – Résultats de la méthode COMET sur la fonction quadratique

Fonction en forme de fauteuil En dimension 2, la fonction est de la forme suivante :

$$f(x, y) = x^4 + y^3$$

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 0.5\}$, ce qui correspond à la situation de la Figure IV.43 à gauche. Les solutions forment une composante connexe (d'un seul tenant, sans trou) continue, légèrement excentrée. L'ordre de grandeur de $f(x, y)$ est 1.5. On cherche à obtenir 500 points dans une zone de tolérance égale à 1% de l'ordre de grandeur. On obtient les points solutions en rouge sur la Figure IV.43 à droite. Là encore, on obtient des points partout autour de la solution, ce qui montre la bonne répartition des points solutions.

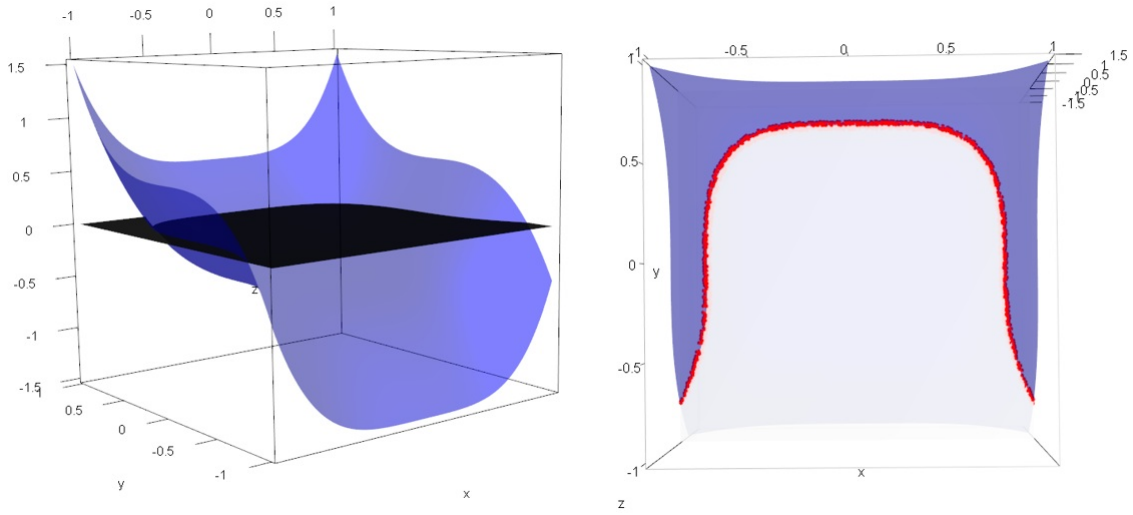

Figure IV.43 – Application de la méthode COMET sur la fonction en forme de fauteuil

Fonction de Rosenbrock En dimension 2, la fonction est de la forme suivante :

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 50\}$, ce qui correspond à la situation de la Figure IV.44 à gauche. Les solutions forment deux courbes bien séparées et très régulières. Le risque est de n'obtenir des points que sur une des deux courbes. L'ordre de grandeur de $f(x, y)$ est 300. On cherche à obtenir 1000 points dans une zone de tolérance égale à 1% de l'ordre de grandeur. On obtient les points solutions en rouge sur la Figure IV.44 à droite. On trouve des solutions bien réparties sur les deux courbes.

Fonction d'Himmelblau En dimension 2, la fonction est de la forme suivante :

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 400\}$, ce qui correspond à la situation de la Figure IV.45 à gauche. Les solutions forment trois courbes très excentrées dont une très petite. L'ordre de grandeur de $f(x, y)$ est 400. On cherche à obtenir 100 points dans une zone de tolérance égale à 1% de l'ordre de grandeur. On obtient les points solutions en rouge sur la Figure IV.45 à droite. La méthode trouve des points autour des trois courbes solutions, peu importe la taille ou la position des courbes.

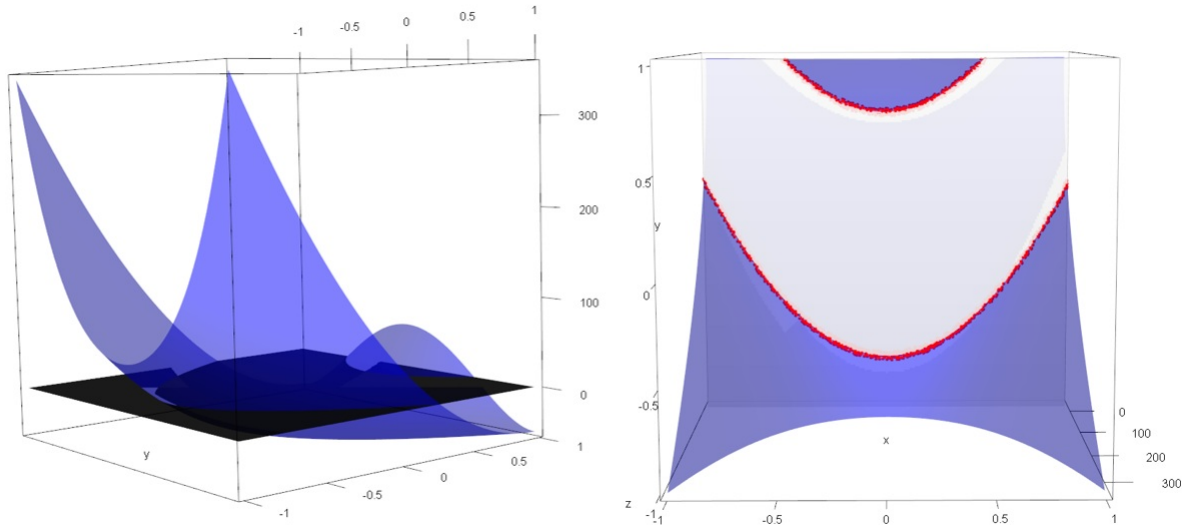

Figure IV.44 – Application de la méthode COMET sur la fonction de Rosenbrock

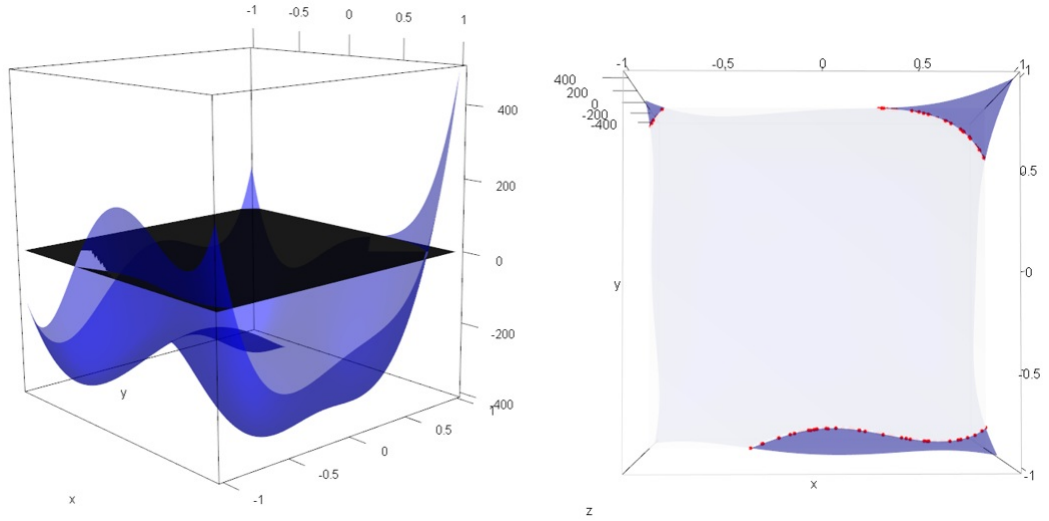

Figure IV.45 – Application de la méthode COMET sur la fonction d'Himmelblau

Fonction polynomiale En dimension 2, on choisit une fonction de la forme :

$$f(x, y) = (x - 0.5)^2 + 3xy - y^3$$

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 2.25\}$, ce qui correspond à la situation de la Figure IV.46 à gauche. La particularité de S est qu'elle contient une partie infinie (la courbe en bas à gauche) et un point en haut à droite du domaine. Ce cas a pour but

de vérifier si la méthode est capable de trouver un point isolé face à une courbe. L'ordre de grandeur de $f(x, y)$ est 4. On cherche à obtenir 100 points dans une zone de tolérance égale à 1% de l'ordre de grandeur. On obtient les points solutions en rouge sur la Figure IV.46 à droite. Le point isolé a été détecté, il y a quelques points solutions, en rouge, proches de ce point.

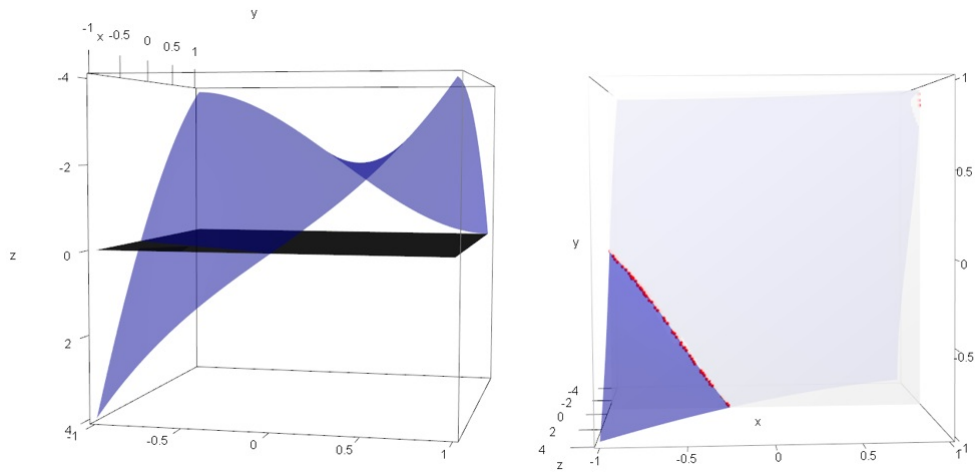

Figure IV.46 – Application de la méthode COMET sur la fonction polynomiale

Fonction trigonométrique En dimension 2, la fonction est de la forme suivante :

$$f(x,y) = 8\sin(7(x-0.9)^2)^2) + 6\sin((14(x-0.9)^2)^2) + (x-0.9)^2 + 8\sin((7(y-0.9)^2)^2) + 6\sin((14(y-0.9)^2)^2) + (y-0.9)^2$$

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 15\}$, ce qui correspond à la situation de la Figure IV.47 à gauche. Les solutions sont un amas de petites composantes connexes assez irrégulières. L'ordre de grandeur de $f(x, y)$ est 12. On cherche à obtenir 5000 points dans une zone de tolérance égale à 10% de l'ordre de grandeur. Nous sommes obligés de demander un grand nombre de points solutions pour représenter toutes ces composantes connexes. On obtient les points solutions en rouge sur la Figure IV.47 à droite. Les points obtenus retracent bien toutes les composantes irrégulières.

Fonction de Rastrigin En dimension 2, la fonction est de la forme suivante :

$$f(x,y) = 20 + x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y) \tag{IV.38}$$

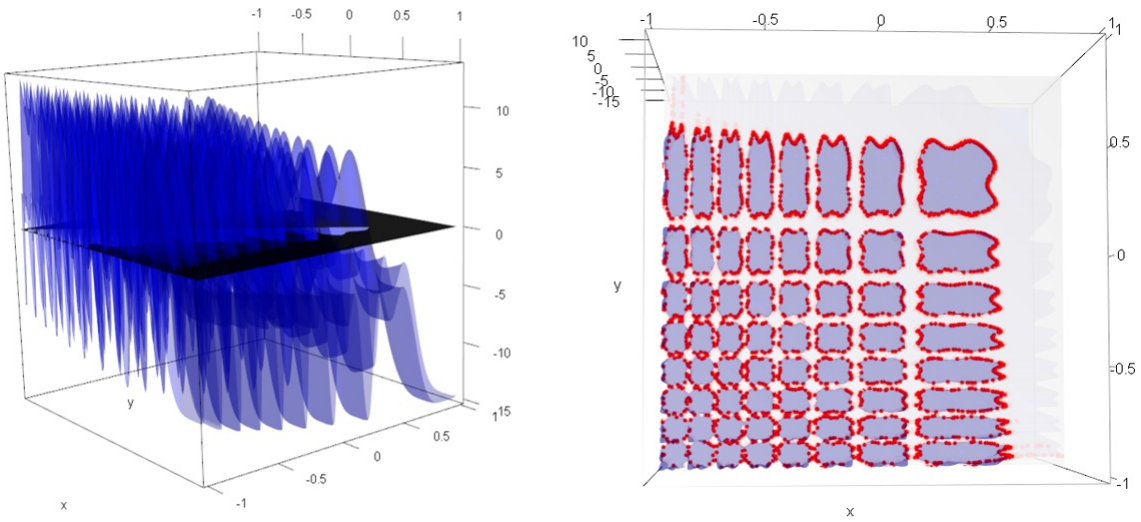

Figure IV.47 – Application de la méthode COMET sur la fonction trigonométrique

On veut obtenir des points proches de la solution $S = \{(x, y) : f(x, y) = 60\}$, ce qui correspond à la situation de la Figure IV.48 à gauche. Les solutions forment un ensemble de petites composantes connexes circulaires, très excentrées. L'ordre de grandeur de $f(x, y)$ est 40. On cherche à obtenir 600 points dans une zone de tolérance égale à 1% de l'ordre de grandeur. On obtient les points solutions en rouge sur la Figure IV.48 à droite. Là encore, on obtient des points proches de toutes les composantes connexes.

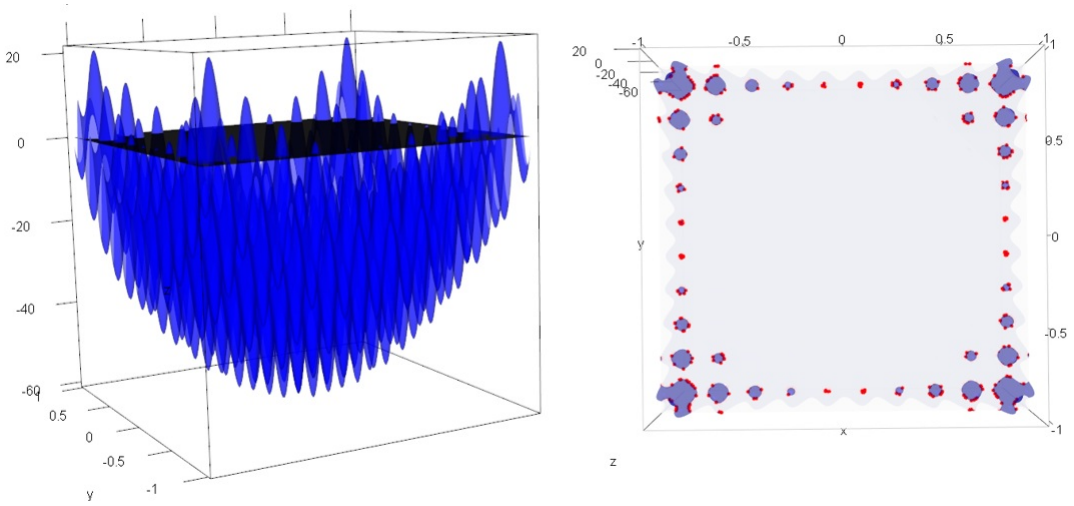

Figure IV.48 – Application de la méthode COMET sur la fonction de Rastrigin

Afin de comparer les temps de calculs et l'efficacité de la méthode suivant la complexité de

Fonction	Type de solution	Temps de calculs	CE
Quadratique	Cercle centré sur le domaine	2.6 s	1.6
Fauteuil	Continue, légèrement excentrée	3.07 s	1.78
Rosenbrock	Deux courbes séparées	4.28 s	1.99
Himmelblau	Trois courbes aux bords du domaine	57.58 s	6.83
Polynomiale	Une courbe et un point au bord du domaine	11.24 s	3.07
Trigonométrique	Plusieurs composantes connexes irrégulières	33.86 s	5.2
Rastrigin	Plusieurs composantes connexes aux bords du domaine	48.61 s	6.4

Tableau IV.10 – Test de la méthodes COMET sur sept fonctions

la fonction et la régularité de la solution (taille, nombre de composantes connexes, etc), nous comparons les résultats de la méthode pour chacun de ces exemples avec la même tolérance (1% de l'ordre de grandeur) et le même nombre de points solutions (100 points). Pour nous ramener à la même tolérance pour toutes les fonctions, on divise chaque fonction par son ordre de grandeur. Ceci fait que chaque fonction a un ordre de grandeur de 1. Les résultats sont répertoriés dans le Tableau IV.10. Les temps de calculs sont exprimés en secondes et le coefficient d'efficacité (CE) est le nombre d'appels à la fonction nécessaire pour obtenir un point solution, en comptant les points initiaux. Tous les résultats sont des moyennes sur 10 lancements de l'algorithme. Les temps de calculs sont liés à l'efficacité de l'algorithme mais aussi à la fonction elle-même. Ainsi, plus la solution à trouver est irrégulière et comporte d'éléments disjoints, plus il faudra d'appels à la fonction pour obtenir un point solution. Plus la fonction est complexe, plus l'algorithme a des difficultés à trouver de nouveaux points.
Les temps de calculs et l'efficacité dépendent également de la dimension du problème et de la tolérance choisie, comme pour la méthode MRM.

4.4 Version finale de la méthode COMET : amélioration et accélération de la méthode modifiée

On définit $\hat{f} : \mathbb{R}^d \to \mathbb{R}$ une fonction permettant d'estimer localement f. Pour un point x, $f(x)$ sera estimé grâce aux valeurs de f sur les k plus proches voisins de x. La définition fournie dans (Burba et al., 2008) est la suivante.

Definition 2 (Noyau)**.** *On appelle noyau la fonction* $K : \mathbb{R}^d \to \mathbb{R}$ *telle que* $\int K = 1$ *et* $\int K^2 < \infty$*. Il s'agit d'une densité de probabilité.*

Definition 3 (Estimateur à noyau des k plus proches voisins)**.** *Soient* $(X_i, Y_i)_{i=1,\dots,n}$ *n paires indépendantes et identiquement distribuées comme* (X, Y) *et à valeurs dans* $D \times \mathbb{R}$*, avec* $(D, \|.\|)$ *un espace métrique. Pour tout* $i \in [\![1, n]\!]$*,* $Y_i = f(X_i)$*. Soit* $x \in D$ *fixé. Les k plus proches voisins de* x *sont les variables* X_i *les plus proches de* x *au sens de la métrique* $\|.\|$*. On définit la fenêtre* h *comme*

$$h := h_{n,k}(x) = \min\{r \in \mathbb{R}^+, \sum_{i=1}^{n} \mathbf{1}_{B(x,r)}(X_i) = k\}, \tag{IV.39}$$

où $\mathbf{1}_{B(x,r)}(X_i)$ *désigne la fonction indicatrice sur* $B(x, r)$*, la boule de centre* x *et de rayon* r*. L'estimateur à noyau des k plus proches voisins de* f *en* x*, noté* $\hat{f}(x)$*, est :*

$$\hat{f}(x) = \sum_{i=1}^{n} Y_i \omega_{i,n}(x), \tag{IV.40}$$

où

$$\omega_{i,n}(x) = \frac{K(\frac{\|x-X_i\|}{h})}{\sum_{i=1}^{n} K(\frac{\|x-X_i\|}{h})}, \tag{IV.41}$$

où K *est un noyau.*

Dans l'algorithme 4 qui sera présenté ci-dessous, les paramètres de l'estimateur des plus proches voisins sont fixés à des valeurs par défaut :

- K est un noyau gaussien,
- k vaut 10,
- h est la distance entre le point étudié x et son pième plus proche voisin parmi les points présents sur le domaine.

Mais l'utilisateur a la possibilité de les modifier en entrée de l'algorithme décrit ci-dessous.

Choix des paramètres
$n \in \mathbb{N}^*$: nombre de points d'initialisation de l'algorithme.
$m << n$: nombre de points intermédiaires.
$q \in]0, 1]$: valeur fixe dépendant du temps de calcul moyen d'une évaluation de f.
tol la tolérance, c'est-à-dire la précision des résultats.
n_f le nombre de points solutions contenus dans la zone de tolérance.
Initialisation
Poser $j = 0$.
Simuler n points $x_1, ..., x_n$ suivant une loi uniforme sur $D \subset \mathbb{R}^d$.
Définir $V_j := \{x_1, \ldots, x_n\}$.
Pour tout $1 \leq i \leq n, y_i := f(x_i)$.
Tant que (le critère d'arrêt n'est pas atteint) **faire**
Poser $j \leftarrow j + 1$.
Ordonner les $y_i : \min_i y_i = y_{(1,n)} \leq y_{(2,n)} \leq \ldots \leq y_{(n,n)} = \max_i y_i$.
Calculer le seuil $S_j = y_{(\lceil nq \rceil, n)}$, quantile d'ordre q des y_i, où $\lceil . \rceil$ désigne la partie entière supérieure.
Simuler un point z suivant une loi uniforme sur D.
Calculer $\hat{f}(z)$ à partir des points de V_j.
Si $(\hat{f}(z) < S_j)$ **Alors**
Conserver z en posant $x_{n+l} = z$, évaluer $f(x_{n+1})$ et poser $y_{n+1} = \hat{f}(z)$.
Sinon
Reprendre à l'étape de la simulation de z.
Fin Si
Poser $n \leftarrow n + 1$.
Simuler m points intermédiaires $u_1, ..., u_m$ suivant une loi uniforme sur D puis évaluer $f(u_1), ..., f(u_m)$.
Mettre à jour V_j : $V_{j+1} := \{x_1, \ldots, x_n\} \cup \{u_1, \ldots, u_m\}$.
Fait
Conclusion
Retourner les points dans la zone de tolérance.

Algorithme 4 – Algorithme modifié selon une méthode à seuil

La méthode est basée sur une sélection par seuil, quantité calculée à chaque itération comme étant le quantile d'ordre q (fixé) des valeurs de $\hat{f}$ aux points simulés. A chaque itération, le point candidat n'est conservé que s'il satisfait le critère de seuil. Si c'est la cas, alors le point est utiliser pour mettre à jour la valeur du seuil.
Les points intermédiaires ne sont pas considérés dans le calcul du quantile mais sont pris en compte pour évaluer $\hat{f}(z)$. Ils permettent d'aller explorer d'autres voisinages du domaine.
Le critère d'arrêt peut correspondre au fait que la suite des quantiles entre dans une zone de tolérance faible autour de la solution. En pratique, on choisira d'obtenir un nombre fini de points dans une zone de tolérance fixée.

L'algorithme peut être résumé graphiquement par la Figure IV.49. Les points initiaux sont

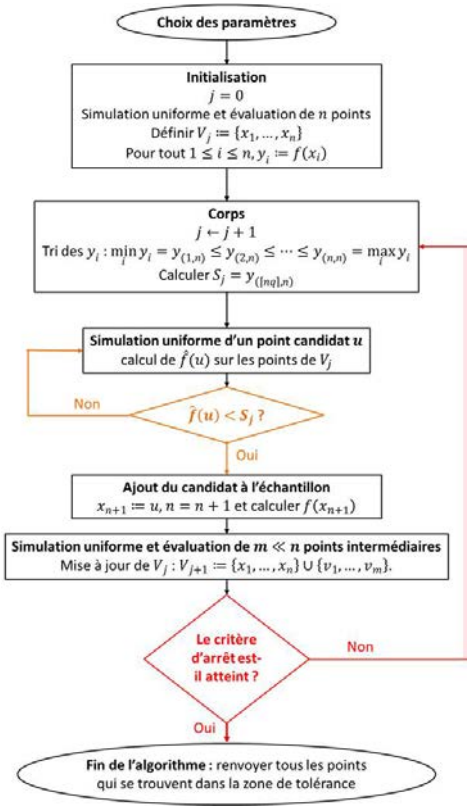

Figure IV.49 – Algorithme COMET modifié selon une méthode à seuil

simulés suivant un plan latin hypercube. Ce type de plan assure l'indépendance et la bonne répartition des points sur le domaine.

La tolérance correspond à la précision des résultats. Pour une tolérance tol et une valeur cible a d'une sortie d'intérêt, les points solutions seront situés dans $[a - tol, a + tol]$.

Le nombre de points finaux correspond au nombre de points dont la sortie est dans l'intervalle de tolérance.

Les points intermédiaires sont des points supplémentaires permettant d'améliorer les estimations locales.

Le choix de q dépend du temps d'évaluation de la fonction étudiée et de la tolérance choisie. Plus il est petit, plus le critère est difficile à satisfaire, plus l'algorithme peut mettre de temps à trouver les points z, moins il y aura d'appels à la fonction f. Plus q est grand, plus les premiers points seront éloignés de la solution, ce qui permet d'aller chercher des solutions partout sur le domaine, ce qui est utile dans les cas où les solutions forment des ensembles disjoints.

Nous comparons cette méthode modifiée à la méthode initiale sur les fonctions usuelles avec les

	Méthode initiale		Méthode modifiée avec $q = 0.25$	
Fonction	**Temps de calculs**	**CE**	**Temps de calculs**	**CE**
Quadratique	2.6 s	1.6	1.14 s	2.08
Fauteuil	3.07 s	1.78	1.13 s	2.11
Rosenbrock	4.28 s	1.99	1.06 s	1.82
Himmelblau	57.58 s	6.83	5.66 s	3.78
Polynomiale	11.24 s	3.07	1.91 s	2.68
Trigonométrique	33.86 s	5.2	30.94 s	4.76
Rastrigin	48.61 s	6.4	38.15 s	5.16

Tableau IV.11 – Comparaison des deux versions de la méthode COMET sur sept fonctions

mêmes réglages de départ. Les résultats sont regroupés dans le Tableau IV.11. Pour la méthode modifiée, on a choisi un ordre de quantile $q = 0.25$. Les résultats obtenus sont des moyennes sur 10 lancements de chaque algorithme dont l'objectif est d'obtenir 100 points dans une zone de tolérance représentant 1% de l'ordre de grandeur de chaque fonction. Les temps de calculs sont toujours en secondes. Sur les méthodes les plus régulières, quadratique et celle en forme de fauteuil, le coefficient d'efficacité est un peu plus élevé avec la méthode modifiée, ce qui montre que cet algorithme est moins efficace que celui de la méthode initiale. Pour autant, les temps de calculs diminuent. Nous avons donc globalement accéléré l'algorithme lui-même. Pour les cinq autres fonctions, la méthode modifiée permet d'améliorer à la fois les temps de calculs et l'efficacité de l'algorithme. Nous utiliserons donc cet algorithme modifié pour appliquer la méthode au cas test principal.

En comparaison à la méthode SAFIP, la méthode COMET demande plus de temps pour atteindre les points solutions (le temps de l'algorithme est plus long) mais le coefficient d'efficacité est nettement amélioré (en général divisé par 2). Elle nécessite donc moins d'appels à la fonction, ce qui est préférable lorsqu'on dispose d'un code de calculs nécessitant plusieurs secondes voire plusieurs minutes par évaluation. C'est pourquoi la méthode COMET a été choisie pour l'application sur le cas test.

Avant cela, nous allons vérifier que la méthode peut s'utiliser en toute dimension puis nous étudierons la convergence de la méthode à travers une version modifiée de l'algorithme. Ensuite, nous verrons que la méthode permet également de résoudre des systèmes d'équations et des problèmes d'optimisation.

	$d = 2$		$d = 3$		$d = 4$	
Fonction	**Temps de calculs**	**CE**	**Temps de calculs**	**CE**	**Temps de calculs**	**CE**
f_1	1.14 s	2.08	10.92 s	5.41	42.6 s	8.13
f_2	1.26 s	3.07	5.03 s	4.85	10.95 s	6.19

Tableau IV.12 – Comparaison des résultats selon la dimension du problème

4.5 Test de la méthode améliorée en dimension supérieure à 2

Nous testons la méthode sur deux fonctions :

$$f_1(x) = \sum_{i=1}^{d} x_i^2 - 0.5$$
$$f_2(x) = \max(|x_1|, \ldots, |x_d|) - 0.5$$

On cherche des points de $S_1 = \{x \in D \subset \mathbb{R}^d : f_1(x) = 0\}$ pour la première fonction et de $S_2 = \{x \in D \subset \mathbb{R}^d : f_2(x) = 0\}$ pour la seconde. L'ensemble S_1 est un cercle pour $d = 2$, une sphère pour $d = 3$ et une hypersphère pour $d \geq 4$. L'ensemble S_2 est un carré pour $d = 2$, un cube pour $d = 3$ et un hypercube pour $d \geq 4$.
Le cas $d = 2$ de f_1 a été étudié à la Section 4.3. Les autres cas sont traités ici et les résultats sont répertoriés dans le Tableau IV.12. Les réglages sont les mêmes que pour le Tableau IV.11, à savoir l'obtention de 100 points dans un voisinage représentant 1% de l'ordre de grandeur de la fonction étudiée. Nous comparons ainsi les résultats suivant la dimension du problème. L'ordre du quantile calculé à chaque itération est choisi comme étant $q = 0.25$. Tous les résultats du tableau sont des moyennes sur 10 lancements de l'algorithme. On remarque que les coefficients d'efficacité augmentent avec la dimension, ce qui est normal. Il en est de même pour les temps de calculs.
En dimension 3 ($d = 3$), on peut visualiser les solutions de $f_1(x) = 0$. Il s'agit d'une sphère de centre $(0, 0, 0)$ et de rayon $\sqrt{0.5}$ représentée à la Figure IV.50. On peut également visualiser les solutions de $f_2(x) = 0$ pour $d = 2$ et $d = 3$ respectivement sur les Figures IV.51 et IV.52 où les solutions sont un carré et un cube. La méthode COMET s'applique donc facilement en dimensions supérieures à 2. Lorsqu'on applique la méthode à un cas concret, le principal inconvénient est que le nombre d'appels à la fonction nécessaire pour avoir un point solution dans la zone de tolérance peut augmenter fortement.

La démonstration de la convergence effectuée à la section suivante est valable en toute di-

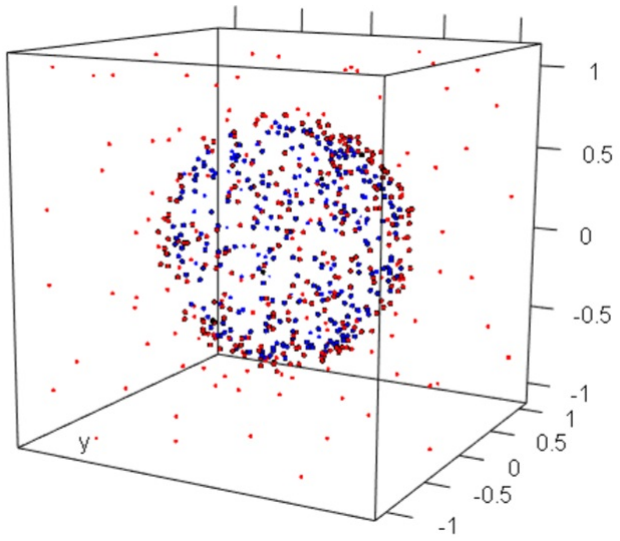

Figure IV.50 – Application de la méthode COMET sur la fonction quadratique en dimension 3

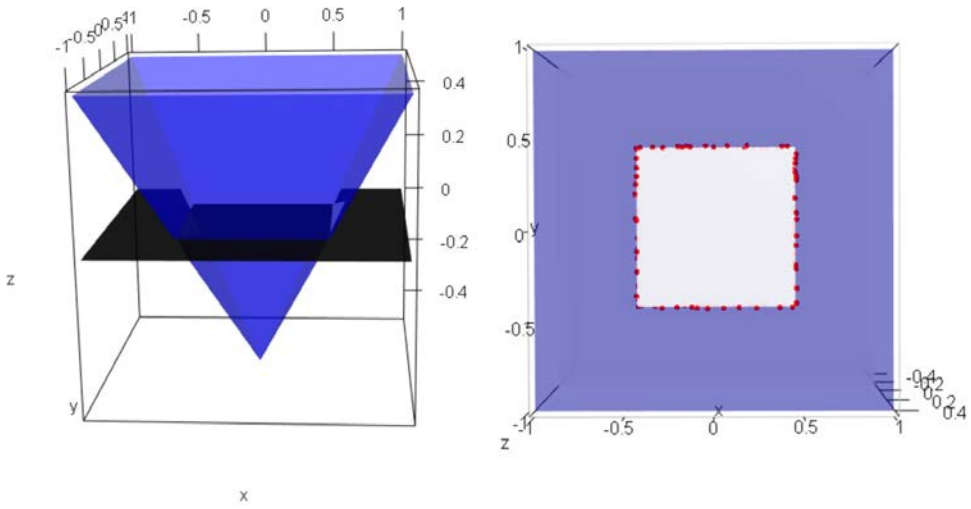

Figure IV.51 – Application de la méthode COMET sur la fonction pyramide en dimension 2

mension.

4.6 Généralisation de la méthode à la résolution de systèmes mal posés

La résolution d'un système consiste à trouver les intersections entre les solutions de chaque équation qui compose le système. Pour généraliser notre méthode à la résolution simultanée de plusieurs problèmes inverses, il suffit de reconsidérer l'étape de sélection des z_l dans l'algorithme

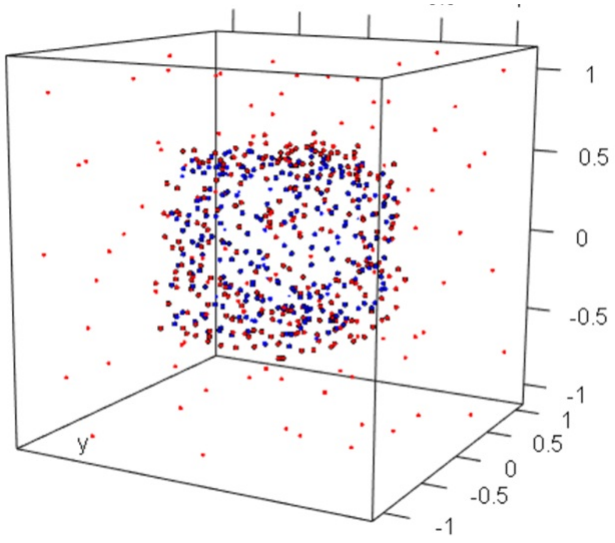

Figure IV.52 – Application de la méthode COMET sur la fonction pyramide en dimension 3

de la Figure 4. Pour chaque fonction du système, une suite de seuils (S_j) est créée. La sélection d'un nouveau point z se fera sous la condition que l'estimateur de f en ce point soit inférieur au seuil relatif à chaque fonction. Une zone de tolérance doit également être choisie pour chaque fonction. Le critère d'arrêt peut alors être le fait d'avoir un certain nombre de points situés dans les deux zones de tolérance simultanément.

Nous avons testé la méthode sur un système composé de deux équations à deux inconnues :

$$\begin{cases} x^2 + y^2 = 0.5, \\ (x - 0.2)^2 + (y + 0.2)^2 = 0.5. \end{cases}$$

Les solutions de chaque équations sont des cercles. Les solutions du système sont donc les intersections de ces deux cercles. Il s'agit ici de deux points.
Nous appliquons la méthode des contours avec $q = 0.25$ et nous choisissons d'obtenir 10 points dans une zone de tolérance à 10^{-2} pour chaque équation. En 5.8 secondes et avec un coefficient d'efficacité de 15.7, nous obtenons des points autour des deux solutions comme nous pouvons le voir sur la Figure IV.53.
Le cercle rouge représente les solutions de la première équation, le cercle bleu celles de la seconde. Les points noirs sont les points solutions obtenus avec la méthode COMET.

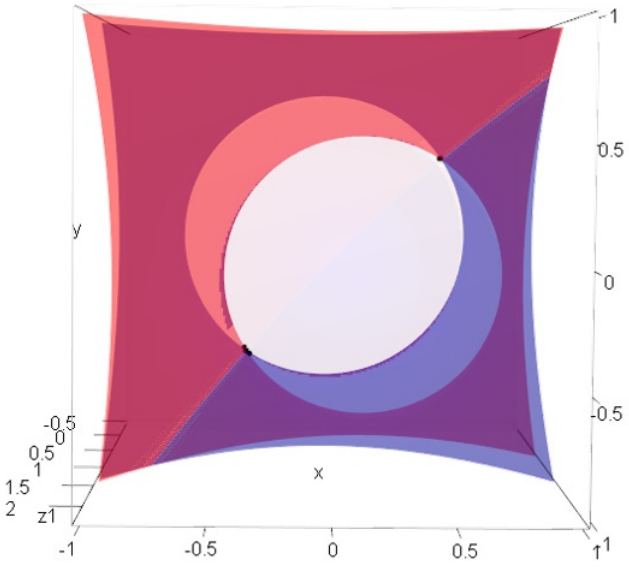

Figure IV.53 – Application de la méthode COMET pour la résolution d'un système de deux équations

5 Comparaison de la méthode MRM et de la méthode COMET

La méthode MRM et la méthode COMET permettent toutes les deux d'obtenir un grand nombre de points solutions sur tout le domaine de variation des entrées. Pour une sortie scalaire, les deux méthodes sont généralisables en toute dimension. Elles permettent également, dans le cas où la solution est régulière, d'en obtenir une expression analytique.
La première différence est que la méthode MRM nécessite une hypothèse forte de monotonie sur la fonction étudiée alors que la méthode COMET ne nécessite aucune hypothèse forte.
Sur une fonction globalement monotone, la méthode MRM sera plus efficace que la méthode COMET. Par contre, sur des fonctions plus génériques, ce n'est pas forcément le cas. Reprenons les exemples tests utilisés dans la section précédente. La méthode des contours permet d'obtenir des points sans pré-traitement sur la fonction, puisqu'il n'y a pas d'hypothèse. Pour appliquer la MRM, il faut s'assurer d'avoir une fonction globalement monotone, ce qui n'est clairement pas le cas de nos sept fonctions usuelles. Il faut donc effectuer un pré-traitement consistant à déterminer les zones de monotonie, sous la condition que la fonction soit différentiable. Comme on est en dimension 2, il ne peut y avoir que quatre types de zones :

1. En vert : $\frac{\partial f}{\partial x} \leq 0$ et $\frac{\partial f}{\partial y} > 0$.

2. En rouge : $\frac{\partial f}{\partial x} > 0$ et $\frac{\partial f}{\partial y} > 0$.
3. En bleu : $\frac{\partial f}{\partial x} \leq 0$ et $\frac{\partial f}{\partial y} \leq 0$.
4. En orange : $\frac{\partial f}{\partial x} > 0$ et $\frac{\partial f}{\partial y} \leq 0$.

La recherche des zones de monotonie a été effectuée sur les sept fonctions usuelles. Les résultats sont fournis à la Figure IV.54. En pratique, l'étude consiste à calculer les dérivées partielles de la fonction et de trouver les changements de signe de ces dérivées. Les figures que nous présentons ici ont été obtenues en quadrillant finement le domaine et en évaluant les dérivées partielles en chaque point du quadrillage. Ceci n'a qu'une valeur illustrative et ne doit pas être utilisée comme méthode de traitement de la monotonie puisqu'elle s'avèrerait très coûteuse.

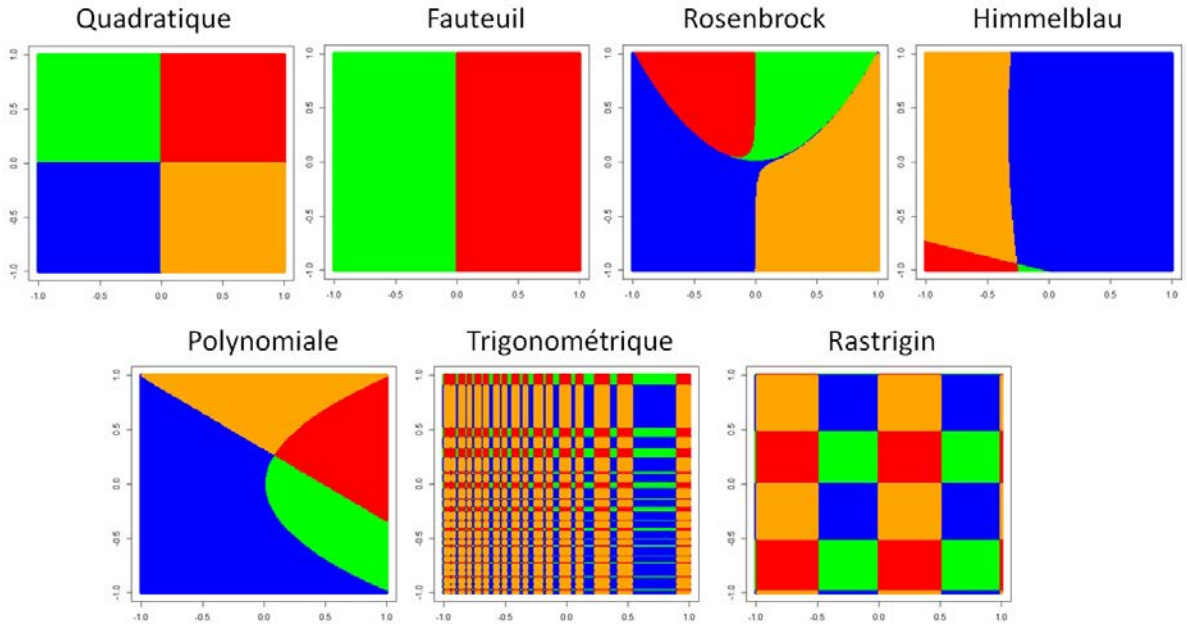

Figure IV.54 – Traitement de la monotonie des sept fonctions jouets pour la méthode MRM

Pour la fonction quadratique et celle en forme de fauteuil, l'étude de la monotonie est assez simple et fait apparaître des zones régulières de monotonie (quatre pour la fonction quadratique et deux pour la fonction en forme de fauteuil). Nous pourrions donc appliquer la méthode MRM sur chacun de ces sous-domaines. Pour la fonction trigonométrique et celle de Rastrigin, les zones semblent également régulières mais sont beaucoup plus nombreuses. Il devient alors compliqué d'appliquer la MRM sur autant de petits sous-domaines dont la plupart risquent d'ailleurs de ne comporter aucune solution. Les trois dernières fonctions présentent des zones de monotonie non rectangles. Ceci pose un sérieux problème à l'utilisation de la méthode MRM. En effet, la méthode est basée sur la simulation aléatoire de points, ce qui n'est pas facile à faire

sur des domaines non rectangles. Ceci pourrait éventuellement fonctionner avec la méthode des segments sur chaque sous-domaine, à partir de points initiaux dans le sous-domaine étudié. Pour obtenir ces points initiaux, il faudrait considérer un rectangle contenant le sous-domaine sur lequel simuler aléatoirement. Les dérivées partielles de la fonction sont calculées en ces points uniformes. Seuls les points satisfaisant les signes des dérivées sur le sous-domaine étudié seront conservés. Cela pourrait effectivement fonctionner mais rend la méthode fastidieuse.

La méthode COMET permet donc des applications plus générales que la méthode MRM. L'absence d'hypothèse sur la fonction étudiée permet même de travailler directement sur le code de calculs, tandis que la méthode MRM nécessite l'utilisation d'un méta-modèle, indispensable pour le traitement de la monotonie, ou d'une hypothèse de monotonie sur le code.

Une autre différence, liée à la première, concerne les temps de calculs et l'efficacité des méthodes. La méthode MRM ne traite que des fonctions globalement monotones, dont le comportement (régularité, forme de l'ensemble de niveau) est assez similaire d'une fonction à l'autre. Les temps de calculs et l'efficacité dépendent uniquement de la dimension du problème et de la précision (tolérance) choisie. La méthode COMET s'applique à des fonctions plus générales dont les comportements peuvent être totalement différents. En plus de la précision et de la dimension, les temps de calculs et l'efficacité dépendent également du comportement de la fonction étudiée, notamment à travers le type de solution recherchée (taille, continuité, nombres d'éléments disjoints, régularité de ces éléments, etc).
La troisième différence porte sur la généralisation des deux méthodes. La méthode MRM, déterministe, pourrait éventuellement être utilisée en non-déterministe. Par contre, la résolution de plusieurs problèmes inverses (un système) simultanément semble difficile par construction. La méthode des contours semble facilement généralisable au cas non-déterministe par la prise en compte de l'aléa lors du calcul de $\hat{f}$ avec la méthode de noyau par plus proches voisins. De plus, nous avons vu que la méthode pouvait résoudre plusieurs problèmes inverses simultanément.

Ces deux méthodes sont appliquées sur le cas test principal de résolution d'un problème d'intégration au niveau du CHP.

6 Cas test principal : résolution d'un problème d'intégration avec les deux méthodes développées

Appliquée au dimensionnement en avant-projets, l'inversion de fonction doit permettre de trouver rapidement des combinaisons des données d'entrée fournissant une valeur cible de la variable d'intérêt, le clash entre la veine et le coin droit du palier 3. Cette sortie est la mesure entre le bas de la veine et le coin droit du palier 3. Elle est représentée sur la Figure IV.55. A gauche, la variable est positive, il n'y a pas de collision entre le compresseur et le palier. A droite, elle est négative, il y a donc un problème d'intégration. Parmi les entrées influentes sur

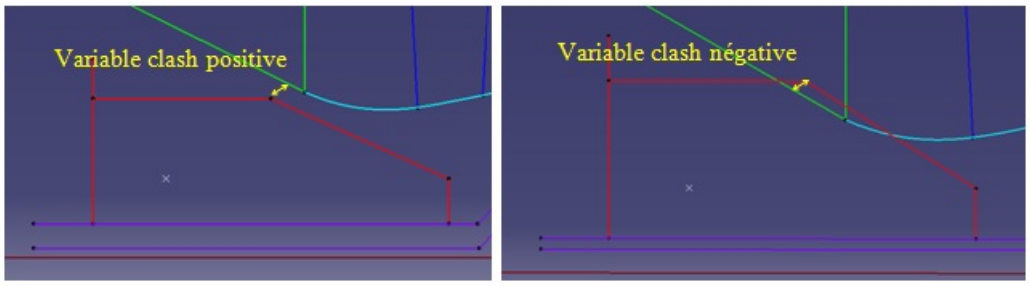

Figure IV.55 – Représentation de la variable d'intérêt, le clash entre la veine et le coin droit du palier 3

la sortie clash, il a été décidé, en accord avec les ingénieurs avant-projets, de ne s'intéresser qu'à deux entrées : F et G. La raison est que l'impact d'un changement de ces deux variables sur le cycle thermodynamique en amont du dimensionnement aéro-mécanique est moindre comparé aux autres variables d'entrée. Les solutions du problème inverse devront également satisfaire toutes les contraintes du système (vecteur Z de sorties) Ces contraintes sont des inégalités). L'optimum en masse trouvé au Chapitre III est considéré comme une contrainte supplémentaire.

Dans un premier temps, nous utiliserons le méta-modèle établi au Chapitre II pour le clash. Nous comparerons les méthodes MRM et COMET sur ce méta-modèle. Puis nous appliquerons la méthode COMET sur le code de calculs. Nous comparerons alors les temps de calculs et le coefficient d'efficacité des méthodes dans ces deux applications.

6.1 Application de la méthode MRM et de la méthode COMET sur le méta-modèle du clash

Dans un premier temps, nous devons définir le problème inverse à résoudre. La variable clash, notée y satisfait $y = f(x)$, où f est le méta-modèle et x est un vecteur regroupant les

entrées F et G. Les autres entrées sont fixées à leur valeur nominale. L'objectif est de définir un dimensionnement satisfaisant toutes les contraintes et pour lequel la variable clash est positive et la plus petite possible. L'idée première est donc de prendre 0 comme valeur cible. Or, comme nous travaillons sur le méta-modèle, nous devons prendre en compte l'erreur de modélisation. En effet, il faut s'assurer que les points solutions obtenus avec les méthodes d'inversion sur le méta-modèle satisfassent bien les deux critères définis précédemment dans la réalité. C'est pour cela que nous choisissons comme cible la moyenne des résidus du modèle à laquelle nous ajoutons deux écarts-types, ce qui nous donne 2.6 mm. Nous cherchons donc l'intersection entre le méta-modèle et le plan horizontal $z = 2.6$ comme le montre la Figure IV.56. Cette figure montre que la solution est continue, il s'agit d'une courbe. Avec le choix d'une tolérance de 0.5 mm, les

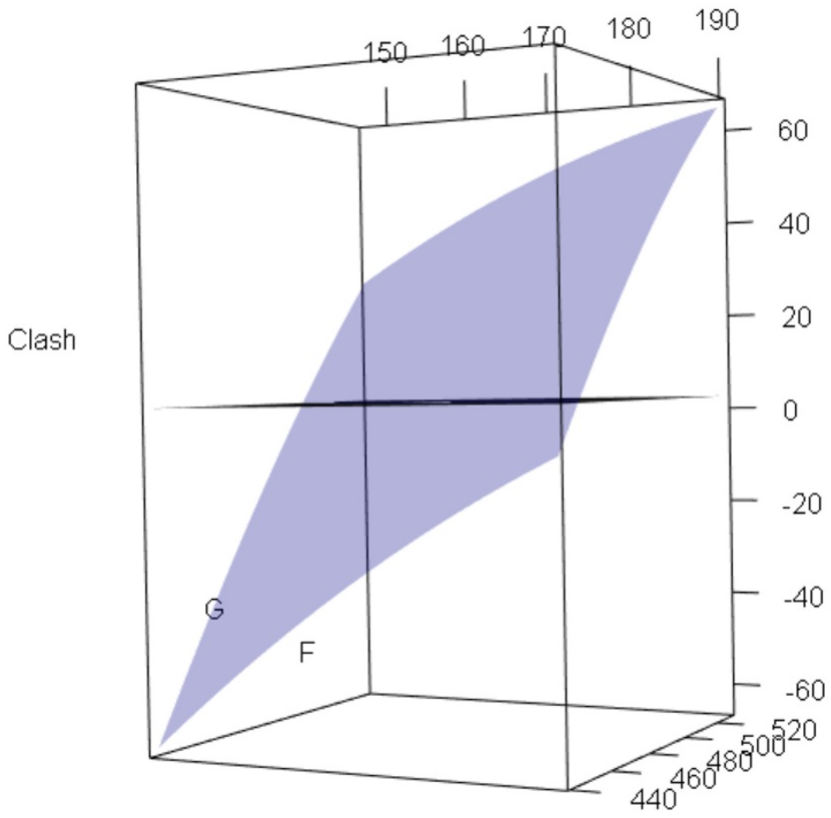

Figure IV.56 – Représentation du méta-modèle du clash en fonction de F et G et tracé du plan correspondant à la cible

points solutions seront donc ceux appartenant à l'ensemble $\tilde{S} = \{x \in D : 2.1 \leq f(x) \leq 3.1\}$.

6.1.1 Résultats de la méthode MRM

Le problème ainsi défini, nous allons d'abord appliquer la méthode MRM. Pour cela, nous avons vérifié la monotonie de la fonction. En F et G, la fonction définie au Chapitre II est quadratique. Sur les intervalles de variation des deux entrées, les deux dérivées partielles du méta-modèle sont positives. Ceci assure la croissance globale de f. Nous pouvons donc appliquer la méthode MRM à cette fonction sans découper le domaine.
Nous choisissons d'obtenir 100 points dans une zone de tolérance à 0.5 mm de la cible 2.6 mm. Avec la méthode des barycentres, ces 100 points sont obtenus en moins d'une seconde et avec un coefficient d'efficacité de 2, ce qui est très bon. En effet, seulement deux appels à la fonction sont nécessaires pour obtenir un point solution. Le problème est que nous n'avons aucune information sur la satisfaction des contraintes par ces 100 points. Il faut donc les évaluer dans le code de calculs. Pour cela, il ne faut pas oublier de dénormaliser les entrées.
L'évaluation d'un point par le code nécessite environ 2 minutes. Ici, il a fallu 2 heures 30 pour évaluer les 100 points. Sur ces 100 points, seuls 20 satisfont les contraintes et sont donc considérés comme des solutions. De plus, comme notre méta-modèle est un bon ajustement du code de calculs, les vraies valeurs du clash pour les combinaisons trouvées sont proches des valeurs estimées. Ces solutions peuvent donc être proposées aux ingénieurs aérodynamiciens afin qu'ils relancent le dimensionnement de la veine. Cette veine conduira ensuite au dimensionnement mécanique du compresseur qui satisfera donc les contraintes et en particulier le clash entre le palier 3 et la veine ainsi que la contrainte de masse.
En comptant les 5 heures de calculs du plan d'expériences utilisé pour la construction du méta-modèle, la méthode MRM a nécessité 8 heures de calculs et 1 à 3 heures de temps humain pour le méta-modèle et l'étude de monotonie.

6.1.2 Résultats de la méthode COMET

Nous comparons l'application de la méthode MRM à celle de la méthode COMET. Nous conservons donc les mêmes objectifs avec la même cible et le même nombre de points exigé dans la même zone de tolérance. Nous choisissons l'algorithme de la Figure 4 qui donne de meilleurs résultats et est plus simple à implémenter que l'algorithme de la Figure IV.40. Nous choisissons $q = 0.25$. En 7 secondes, la méthode a permis d'obtenir les 100 points dans la zone de tolérance souhaitée, avec un coefficient d'efficacité de 1.94, soit légèrement meilleur que la méthode MRM. Notons que la méthode initiale, utilisant l'algorithme de la Figure IV.40, avait nécessité 22 secondes de calculs et présentait un coefficient d'efficacité de 11.34. En 2 heures 28, ces 100 points sont évalués dans le code et nous obtenons 17 points satisfaisant les contraintes.

Les deux méthodes, appliquées sur un cas test satisfaisant la condition de monotonie globale, sont donc équivalentes. Nous avons vu dans la section précédente que ce n'est plus le cas pour des fonctions plus complexes et moins régulières. En effet, la méthode COMET permet facilement des applications plus générales. Cette généralité permet d'ailleurs une utilisation directe sur le code de calculs dont il est question dans la section suivante.

6.2 Application de la méthode COMET directement sur le code de calculs

Le code de calculs s'effectue dans le logiciel Optimus et la méthode COMET a été implémentée dans R. Pour pouvoir appliquer la méthode directement sur le code, un couplage entre les deux logiciels est nécessaire. Grâce à ce couplage, on obtient, à chaque calcul, la vraie valeur du clash et toutes les contraintes.
Comme nous voulons obtenir des points avec un clash positif faible, nous choisissons une cible à 0.5 mm et une tolérance à 0.5 mm. Les points solutions appartiennent donc à $\tilde{S} = \{x \in D : 0 \leq x \leq 1\}$. Nous voulons obtenir 10 points dans cet ensemble qui satisfont toutes les contraintes. Nous conservons $q = 0.25$. La méthode converge en 3 heures 35 avec un coefficient d'efficacité de 11. Ce coefficient est élevé mais ceci est principalement dû au fait que le système est très contraint. A titre de comparaison, la méthode COMET avec l'algorithme de la Figure IV.40 a convergé en 14 heures 24 avec un coefficient d'efficacité de 69. L'algorithme de la Figure 4 est donc beaucoup plus efficace et plus rapide que la méthode initiale.
La Figure IV.57 montre les résultats de la méthode d'inversion. On y représente les points de l'algorithme dans le plan formé par les deux variables d'entrée normalisées. Les points sont donc représentés dans le plan $[-1, 1]^2$. En rouge et en bleu, ce sont les différents points évalués par l'algorithme. En noir, ce sont tous les points situés dans la zone de tolérance choisie. En vert, ce sont les points dans la zone de tolérance qui satisfont les contraintes. Ces points verts sont donc nos points solutions. Le fait que les points (en noir) forment une droite est uniquement dû à la régularité de la sortie clash par rapport aux deux variables d'entrée. Nous remarquons également que les points solutions (en vert) sont assez bien répartis (ils ne sont pas tous regroupés au même endroit). Ceci est très important car cela permet de fournir des solutions différentes. Comme la méthode ne nécessite aucun pré-traitement, notamment pas de méta-modèle, le temps total pour obtenir 10 points est de 3 heures 35. L'application sur le méta-modèle a nécessité beaucoup plus de temps, en partie à cause du pré-traitement mais aussi par rapport à l'obligation d'évaluer les points solutions. La sévérité des contraintes du système fait que nous ne savons pas combien de points nous obtiendrons finalement. Il est donc

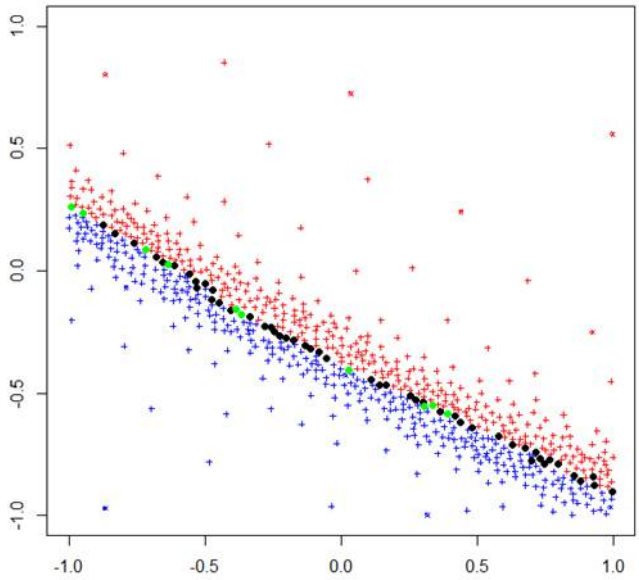

Figure IV.57 – Résultats de la méthode d'inversion, sous contraintes, pour le clash entre le palier 3 et la veine, obtention de 10 points

préférable d'en chercher beaucoup avec la méthode COMET. Avec l'application sur le code, ce problème ne se pose plus, nous obtenons exactement le nombre de points désirés, tous satisfont les contraintes.
Les 10 points solutions sont des solutions normalisées. Afin de les fournir aux ingénieurs de l'aérodynamique pour le rebouclage, nous devons les dénormaliser, c'est-à-dire les ré-exprimer dans leur unité physique initiale. Parmi ces propositions, ils pourront choisir celles qui leur convient le mieux en termes de respect des spécifications et de performances. Quel que soit leur choix, il n'y aura plus de problème d'intégration en sortie du dimensionnement mécanique et toutes les autres contraintes seront satisfaites.

Un test a également été effectué pour obtenir 30 points solutions. Les résultats sont fournis à la Figure IV.58. Nous pouvons donc fournir ces résultats en entrée de l'aérodynamique, sans repasser par le code. Avec ce cas de figure, la méthode peut être implémentée au sein même des outils de dimensionnement pour être utilisée de manière interactive. Ainsi, dès qu'un problème du type problème d'intégration ou non satisfaction d'une spécification est détecté, l'outil peut trouver directement des configurations possibles et les tester. Ceci facilite et accélère le dimensionnement.

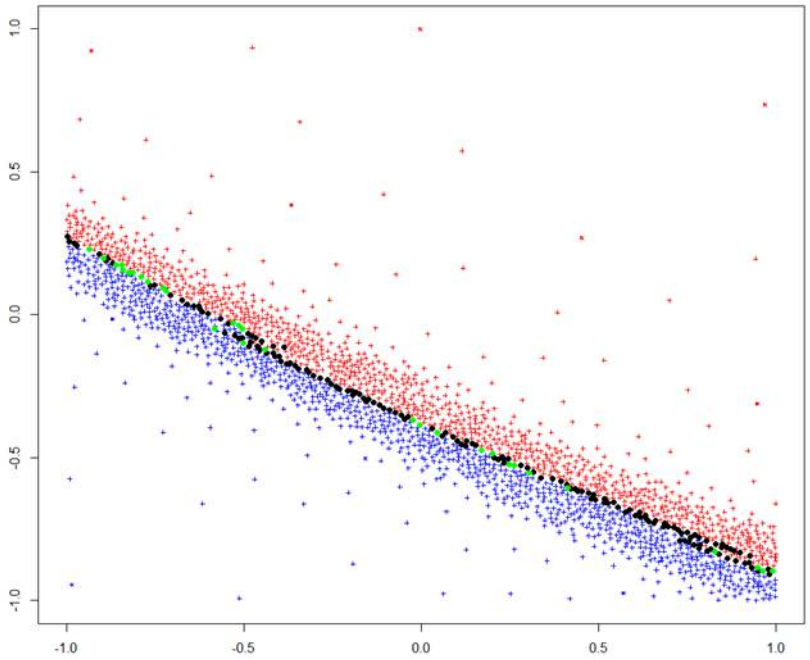

Figure IV.58 – Résultats de la méthode d'inversion, sous contraintes, pour le clash entre le palier 3 et la veine, obtention de 30 points

	MRM			Méthode initiale contours			Méthode modifiée contours		
Fonction	**Temps de calculs**	**Temps humain**	**CE**	**Temps de calculs**	**Temps humain**	**CE**	**Temps de calculs**	**Temps humain**	**CE**
Métamodèle	7h30	1 à 5h	2	7h30	1 à 3h	11.34	7h30	1 à 3h	1.94
Code de calculs	-	-	-	14h24	0	69	3h35	0	11

Tableau IV.13 – Récapitulatif des résultats des méthodes d'inversion sur le cas test principal

6.3 Bilan de l'application des méthodes d'inversion sur le méta-modèle du clash

Les comparaisons des temps de calculs et de l'efficacité des méthodes développées, MRM et deux variantes de COMET, sont répertoriées dans le Tableau IV.13. Dans des cas très réguliers où le méta-modèle s'obtient facilement, nous pouvons utiliser indifféremment les méthodes MRM et COMET. Par contre, si la fonction étudiée devient fortement irrégulière ou que le problème possède de nombreuses contraintes, il est plus efficace de considérer la méthode COMET directement sur le code de calculs.

Nous avons vu que les méthodes développées durant cette thèse, notamment la méthode CO-

MET, s'appliquent bien à des problèmes inverses mal posés. Nous allons voir dans la section suivante que dans des cas plus simples où le problème est bien posé, il est préférable d'utiliser les méthodes usuelles qui sont alors plus efficaces.

7 Cas test secondaire : évaluation d'un effort à partir de signaux de jauges

Lorsque la phase de calibration des jauges est terminée, nous disposons d'un modèle exprimant les déformations en fonction de l'effort, vecteur composé de la force et de l'angle. Il s'agit de la fonction de transfert que nous avons obtenue au Chapitre II. Lors de l'évaluation de l'effort, des mesures de déformations sont effectuées lors d'un essai en rotation. A chaque pas de temps, on dispose de mesures en chaque jauge située sur le palier et on veut retrouver l'effort appliqué. Il s'agit donc d'inverser la fonction de transfert à chaque pas de temps et pour chacune des jauges. Le problème inverse est un système à 4 équations (une pour chaque jauge) et 2 inconnues (la force et l'angle). La force et l'angle sont aussi appelées amplitude et phase de l'effort. La solution existe et est unique. Le problème est donc bien posé.

Dans l'exemple que nous avons traité, l'essai dure 2 minutes. Les déformations mesurées en chaque jauge sont représentées à la Figure IV.59.

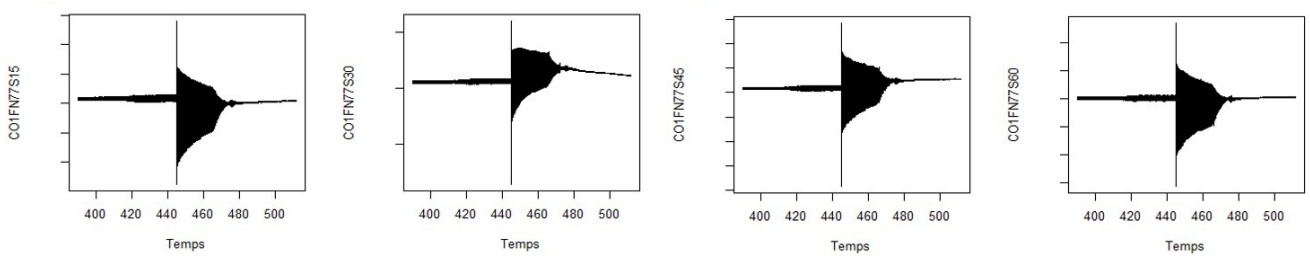

Figure IV.59 – Résultats de l'essai en rotation pour chacune des 4 jauges

En zoomant sur ces données temporelles, on voit apparaître le cycle de rotation du palier. Il s'agit donc d'un signal périodique où chaque période correspond plus ou moins nettement selon l'instant choisi à la fonction de transfert, comme le montre la Figure IV.60.
La partie intéressante sur la Figure IV.59 pour les ingénieurs est celle située au niveau du pic. Ceci correspond à un évènement (ingestion, perte d'aube, etc). Avant le pic, le palier étudié se trouve en fonctionnement normal, sans effort (force et angle nuls). Après, les déformations entrent dans un mouvement oscillatoire (comme représenté à la Figure IV.60) où l'effort appli-

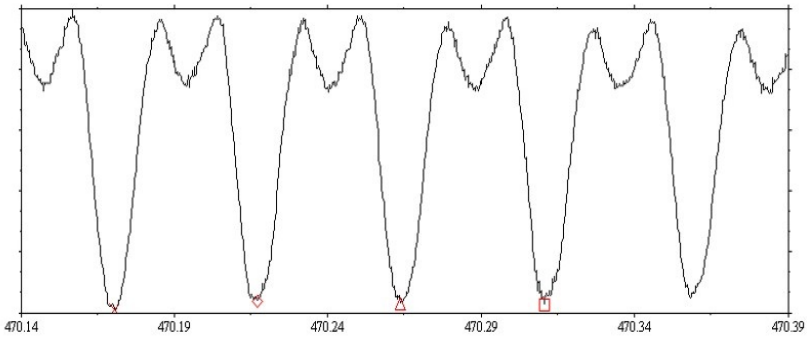

Figure IV.60 – Zoom des résultats de l'essai en rotation sur 0.25 secondes

qué diminue au cours du temps, jusqu'à retrouver le mode de fonctionnement normal.
Nous sélectionnons une plage de mesure (entre 445 et 455 secondes) autour de ce pic. Les données sont ensuite filtrées de deux façons :

- un filtre passe-haut à 10 Hz pour éliminer la composante statique,
- un filtre passe-bas à 3 fois le régime de rotation s'il est connu, sinon à 1250 Hz.

Les données que nous voulons inverser sont représentées à la Figure IV.61. A chaque pas de

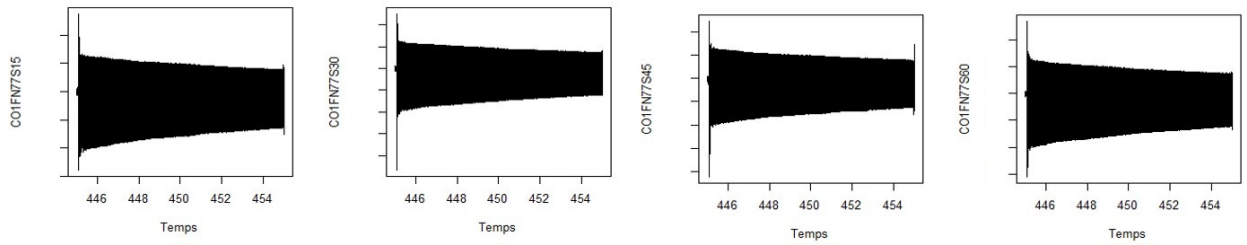

Figure IV.61 – Représentation des données de recombinaison, sélectionnées et filtrées

temps correspond une valeur de déformation, ce qui correspond à la valeur cible pour les quatre équations. La solution recherchée est l'intersection des solutions de chacune des équations. Le problème est que chaque inversion est faite sur un modèle qui a été déterminé sur des valeurs statiques. En passant aux valeurs en rotation, il y a une erreur de modélisation. Cette erreur implique que les quatre inversions ne se croisent pas exactement en un point. Ainsi, la méthode COMET va faire apparaître plusieurs points solutions. De plus, si les croisements sont trop éloignés et que l'on a choisi une tolérance faible, la méthode risque de ne pas converger. Ensuite, la résolution pour un pas de temps peut prendre quelques secondes. Or, nous avons 3125 mesures par secondes, soit autant de systèmes d'équations à résoudre sur la sélection que nous avons faite, entre 445 et 455 secondes. Enfin, la méthode COMET a été conçue pour résoudre des systèmes complexes avec de nombreuses solutions. Il n'est peut-être pas nécessaire de faire appel à une telle méthode lorsque le problème peut être résolu avec des méthodes

usuelles.
Face à cela, nous décidons d'utiliser une méthode usuelle de Newton, appliquée au problème exprimé sous forme des moindres carrés. En chaque jauge, l'erreur s'écrit : $(Y_j - J_j)^2$, où Y_j est l'estimation par la fonction de forme et J_j le signal mesuré en la jauge j. On cherche à minimiser l'erreur totale :

$$E_{tot} = \sum_{j=1}^{4} (Y_j - J_j)^2. \tag{IV.42}$$

Pour un tel problème, la méthode de Newton permet d'obtenir un résultat instantanément. On pourrait appliquer la méthode COMET sur ce problème de moindres carrés mais elle serait moins efficace que la méthode de Newton. Lorsqu'on a quelques milliers de résolutions à effectuer successivement, le moindre dixième de seconde supplémentaire pour une résolution a d'importantes répercussions sur le temps total.
Avec la méthode de Newton, nous devons faire un choix d'initialisation pour le premier pas de temps. Pour les suivants, le point de départ de la résolution est la solution du point précédent. Sur la plage de temps sélectionnée, nous obtenons en 5 minutes les résultats représentés à la Figure IV.62.

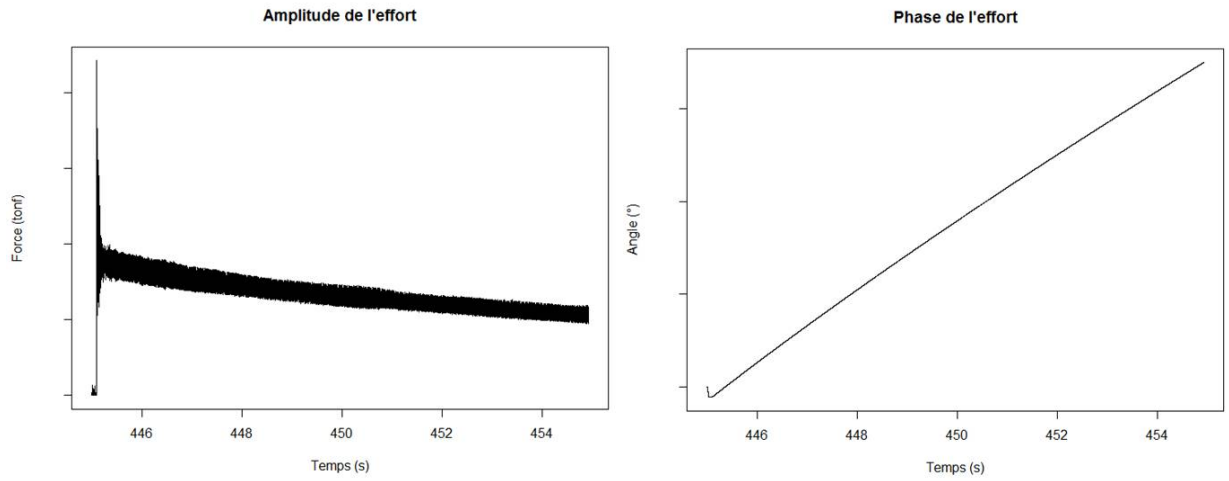

Figure IV.62 – Résultats de l'effort obtenu par la méthode de Newton

Avant le pic, on a bien un effort nul : la force et l'angle sont nuls. La légère oscillation en amplitude provient de l'oscillation présente sur les mesures de déformations. Il s'agit de l'erreur de mesure. Au moment du pic, il y a une forte augmentation de la force. Puis elle décroît au cours du temps. La phase de l'effort (ou l'angle) prend en compte le nombre de tours effectués par le palier depuis le pic, d'où cette allure linéaire croissante.
Les résultats obtenus sont cohérents aux connaissances physiques que nous avons du phéno-

mène. En dernier lieu, nous comparons ces résultats à ceux obtenus avec le modèle de base d'ordre 4, c'est-à-dire sans prise en compte de la force et sans extrapolation. Cette comparaison est représentée à la Figure IV.63. On voit sur cette figure, que le modèle sans extrapolation et

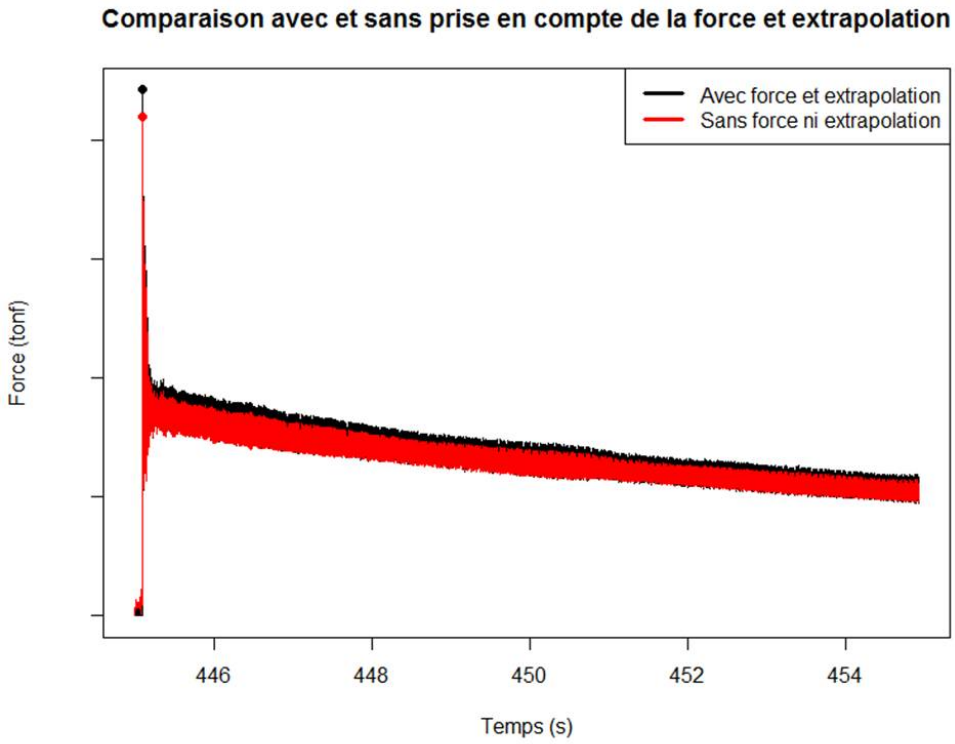

Figure IV.63 – Comparaison des résultats avec deux modèles différents, avec ou sans prise en compte de la force et l'ajout de points d'extrapolation

sans prise en compte de la force, sous-évalue l'amplitude de l'effort. Le décalage absolu moyen est de 0.73 tonf. Sur le pic l'écart est de 4.5 tonf. Le choix du modèle est donc important et doit être de bonne qualité pour assurer de meilleurs résultats lors de l'évaluation de l'effort. La prise en compte de la force dans le modèle et l'extrapolation lors de la modélisation permettent d'améliorer le modèle initial dont l'ordre doit être bien choisi.

8 Conclusion

Dans ce chapitre, nous avons fait un état de l'art des méthodes de résolution de problèmes inverses. De nombreuses méthodes permettent de résoudre des problèmes bien posés. Cependant, le plus souvent, les problèmes inverses sont des problèmes mal posés. L'utilisation des méthodes usuelles nécessite alors une étape de régularisation afin de rendre le problème bien

posé. Cette régularisation consiste à ajouter a priori des contraintes au problème. Ici, nous voulions résoudre directement le problème sans utiliser de connaissances a priori, que nous n'avons d'ailleurs pas forcément.
Nous avons alors cherché des méthodes permettant de résoudre des problèmes mal posés. De telles méthodes existent mais présentent certains inconvénients que nous avons voulu éviter. C'est pourquoi nous avons développé deux nouvelles méthodes. La première, issue de la fiabilité des structures, permet de résoudre, en toute dimension, des problèmes inverses mal posés, sous condition de monotonie. Face à cette hypothèse forte, nous avons développé la méthode COMET. L'absence d'hypothèse forte sur la fonction étudiée permet à la méthode d'être utilisée directement sur les codes de calculs. Équivalente en termes de temps de calculs et d'efficacité à la méthode MRM pour des fonctions monotones, la méthode COMET est également applicable en toute dimension. De plus, elle permet beaucoup plus facilement de résoudre des inversions pour des fonctions très irrégulières, avec des ensembles solutions disjoints. Nous avons d'ailleurs pu le constater en la testant sur des fonctions bien connues, très chahutées pour certaines. Enfin, cette méthode présente l'avantage d'être très malléable. Elle s'adapte notamment à la résolution de systèmes d'équations et à des cas d'optimisation.
Nous avons ensuite utilisé ces deux méthodes pour résoudre un problème d'intégration dans un dimensionnement en avant-projets. Les deux méthodes permettent d'obtenir des solutions qui satisfont toutes les contraintes. Cependant, l'application de la méthode COMET sur le code de calculs permet d'en obtenir beaucoup plus rapidement, notamment car il n'y a alors aucun pré-traitement à effectuer et que les points solutions satisfont directement les contraintes.
Enfin, nous avons vu les limites de nos méthodes avec le cas test secondaire. D'abord, la méthode MRM ne s'applique pas du tout puisqu'elle ne permet pas de résoudre les systèmes d'équations et que la fonction de transfert n'est pas monotone. Ensuite, la méthode COMET peut trouver la solution de chaque système mais les méthodes usuelles telles que la méthode de Newton est suffisante et beaucoup plus rapide. Pour des problèmes bien posés, on peut donc se contenter des méthodes bien connues, habituellement utilisées et implémentées dans les logiciels tels que R ou Matlab.

Chapitre V

Theoretical basis for an inverse method using extreme deviations

Let $X_1, \ldots, X_n$ denote n independent real-valued random variables under distribution P_X. The random walk $(X_1, \ldots, X_n)$ is conditioned on an extreme deviation of its sum $(S_1^n = na_n)$ or $(S_1^n > na_n)$ where $a_n \to \infty$. We set

$$C_a := (\frac{S_1^n}{n} > a), \tag{V.1}$$

and

$$I_a := \bigcap_{i=1}^{n} (X_i > a). \tag{V.2}$$

It is proved that when the summands have light tails with some additional regularity property, then, conditioned on C_a, all observations of X_i will be close to a, with $a \to \infty$ when n tends to infinity. This can be written as

$$\lim_{a \to \infty} \mathbb{P}(I_a | C_a) = 1. \tag{V.3}$$

With some additional conditions on f, we obtain the same results for $f(X_1), \ldots, f(X_n)$. If we are able to simulate $X_1, \ldots, X_n$ (or just X_1) conditioned on $\frac{1}{n}\sum_{i=1}^{n} f(X_i) > a$, then X_1 satisfies $f(X_1) \in (a - \epsilon_n, a + \epsilon_n)$ with ϵ_n small with n and depending on a and the distribution of $f(X)$.

It is proved that when the summands have light tails, we have also that the asymptotic conditional distribution of X_1 can be approximated by the tilted distribution (defined in the chapter

by V.5) at point a_n in variation norm, extending therefore the classical LDP (Large Deviation Principle) case.

This chapter is divided in two sections, each one is an article. In the first article, (Biret et al., 2015), we focus on asymptotic equivalents for the moment of the tilted distribution and on its Gaussian behavior. In the second article, (Biret et al., 2016), we explore a conditional Gibbs theorem for the conditioned distribution of the random walk.

1 A sharp Abelian theorem for the Laplace transform

Abstract This paper states asymptotic equivalents for the moments of the Esscher transform of a distribution on $\mathbb{R}$ with smooth density in the upper tail. As a by product if provides a tail approximation for its moment generating function, and shows that the Esscher transforms have a Gaussian behavior for large values of the parameter.

1.1 Introduction

Let X denote a real-valued random variable with support $\mathbb{R}$ and distribution P_X with density p.

The moment generating function of X

$$\Phi(t) := \mathbb{E}[\exp(tX)] \tag{V.4}$$

is supposed to be finite in a non void neighborhood $\mathcal{N}$ of 0. This hypothesis is usually referred to as a Cramér type condition.

The tilted density of X (or Esscher transform of its distribution) with parameter t in $\mathcal{N}$ is defined on $\mathbb{R}$ by

$$\pi_t(x) := \frac{\exp(tx)}{\Phi(t)} p(x). \tag{V.5}$$

For $t \in \mathcal{N}$, the functions

$$t \to m(t) := \frac{d}{dt} \log \Phi(t), \tag{V.6}$$

$$t \to s^2(t) := \frac{d^2}{dt^2} \log \Phi(t), \tag{V.7}$$

$$t \to \mu_j(t) := \frac{d^j}{dt^j} \log \Phi(t), j \in (2, \infty). \tag{V.8}$$

are respectively the expectation and the centered moments of a random variable with density π_t.

When Φ is steep, meaning that

$$\lim_{t \to t^+} m(t) = \infty \tag{V.9}$$

and

$$\lim_{t \to t^-} m(t) = -\infty$$

where $t^+ := ess\sup \mathcal{N}$ and $t^- := ess\inf \mathcal{N}$ then m parametrizes $\mathbb{R}$ (this is steepness, see (Barndorff-Nielsen, 1978)). We will only require (V.9) to hold.

This paper presents sharp approximations for the moments of the tilted density π_t under conditions pertaining to the shape of p in its upper tail, when t tends to the upper bound of $\mathcal{N}$.

Such expansions are relevant in the context of extreme value theory as well as in approximations of very large deviation probabilities for the empirical mean of independent and identically distributed summands. We refer to (Feigin and Yashchin, 1983) in the first case, where convergence in type to the Gumbel extreme distribution follows from the self neglecting property of the function s^2, and to (Broniatowsk and Mason, 1994) in relation with extreme deviation probabilities. The fact that up to a normalization, and under the natural regularity conditions assumed in this paper, the tilted distribution with density $\pi_t(x)$ converges to a standard Gaussian law as t tends to the essential supremum of the set $\mathcal{N}$ is also of some interest.

1.2 Notation and hypotheses

Thereafter we will use indifferently the notation $f(t) \underset{t\to\infty}{\sim} g(t)$ and $f(t) \underset{t\to\infty}{=} g(t)(1+o(1))$ to specify that f and g are asymptotically equivalent functions.

The density p is assumed to be of the form

$$p(x) = \exp(-(g(x) - q(x))), \quad x \in \mathbb{R}_+. \tag{V.10}$$

For the sake of this paper, only the form of p for positive x matters.

The function g is positive, convex, four times differentiable and satisfies

$$\frac{g(x)}{x} \underset{x\to\infty}{\longrightarrow} \infty. \tag{V.11}$$

Define

$$h(x) := g'(x). \tag{V.12}$$

In the present context, due to (V.68) and the assumed conditions on q to be stated hereunder, $t^+ = +\infty$.

Not all positive convex g's satisfying (V.68) are adapted to our purpose. We follow the line of (Juszczak and Nagaev, 2004) to describe the assumed regularity conditions of h. See also (Balkema et al., 1993) for somehow similar conditions.

We firstly assume that the function h, which is a positive function defined on $\mathbb{R}_+$, is either regularly or rapidly varying in a neighborhood of infinity ; the function h is monotone and, by (V.68), $h(x) \to \infty$ when $x \to \infty$.

The following notation is adopted.

$RV(\alpha)$ designates the class of regularly varying functions of index α defined on $\mathbb{R}_+$,
$\psi(t) := h^{\leftarrow}(t)$ designates the inverse of h. Hence ψ is monotone for large t and $\psi(t) \to \infty$ when $t \to \infty$,
$\sigma^2(x) := 1/h'(x)$,
$\hat{x} := \hat{x}(t) = \psi(t)$,
$\hat{\sigma} := \sigma(\hat{x}) = \sigma(\psi(t))$.

The two cases considered for h, the regularly varying case and the rapidly varying case, are described below. The first one is adapted to regularly varying functions g, whose smoothness is described through the following condition pertaining to h.

***Case* 1** (The Regularly varying case)**.** *It will be assumed that h belongs to the subclass of $RV(\beta)$, $\beta > 0$, with*

$$h(x) = x^\beta l(x),$$

where

$$l(x) = c \exp \int_1^x \frac{\epsilon(u)}{u} du \tag{V.13}$$

for some positive c. We assume that $x \mapsto \epsilon(x)$ is twice differentiable and satisfies

$$\begin{cases} \epsilon(x) \underset{x\to\infty}{=} o(1), \\ x|\epsilon'(x)| \underset{x\to\infty}{=} O(1), \\ x^2|\epsilon^{(2)}(x)| \underset{x\to\infty}{=} O(1). \end{cases} \tag{V.14}$$

It will also be assumed that

$$|h^{(2)}(x)| \in RV(\theta) \tag{V.15}$$

where θ is a real number such that $\theta \leq \beta - 2$.

Remark 2. *Under (V.70), when $\beta \neq 1$ then, under (V.72), $\theta = \beta - 2$. Whereas, when $\beta = 1$ then $\theta \leq \beta - 2$. A sufficient condition for the last assumption (V.72) is that $\epsilon'(t) \in RV(\gamma)$, for some $\gamma < -1$. Also in this case when $\beta = 1$, then $\theta = \beta + \gamma - 1$.*

Example 10 (Weibull density)**.** *Let p be a Weibull density with shape parameter $k > 1$ and scale parameter 1, namely*

$$\begin{aligned} p(x) &= kx^{k-1}\exp(-x^k), \qquad x \geq 0 \\ &= k\exp(-(x^k - (k-1)\log x)). \end{aligned}$$

Take $g(x) = x^k - (k-1)\log x$ and $q(x) = 0$. Then it holds

$$h(x) = kx^{k-1} - \frac{k-1}{x} = x^{k-1}\left(k - \frac{k-1}{x^k}\right).$$

Set $l(x) = k - (k-1)/x^k, x \geq 1$, which verifies

$$l'(x) = \frac{k(k-1)}{x^{k+1}} = \frac{l(x)\epsilon(x)}{x}$$

with

$$\epsilon(x) = \frac{k(k-1)}{kx^k - (k-1)}.$$

Since the function $\epsilon(x)$ satisfies the three conditions in (V.71), then $h(x) \in RV(k-1)$.

***Case* 2** (The Rapidly varying case)**.** *Here we have $h^{\leftarrow}(t) = \psi(t) \in RV(0)$ and*

$$\psi(t) = c\exp\int_1^t \frac{\epsilon(u)}{u}du \tag{V.16}$$

for some positive c, and $t \mapsto \epsilon(t)$ is twice differentiable with

$$\begin{cases} \epsilon(t) \underset{t\to\infty}{=} o(1), \\ \frac{t\epsilon'(t)}{\epsilon(t)} \underset{t\to\infty}{\longrightarrow} 0, \\ \frac{t^2\epsilon^{(2)}(t)}{\epsilon(t)} \underset{t\to\infty}{\longrightarrow} 0. \end{cases} \tag{V.17}$$

Note that these assumptions imply that $\epsilon(t) \in RV(0)$.

Example 11 (A rapidly varying density)**.** *Define p through*

$$p(x) = c\exp(-e^{x-1}), x \geq 0.$$

Then $g(x) = h(x) = e^{x-1}$ and $q(x) = 0$ for all non negative x. We show that $h(x)$ is a rapidly varying function. It holds $\psi(t) = \log t + 1$. Since $\psi'(t) = 1/t$, let $\epsilon(t) = 1/(\log t + 1)$ such that $\psi'(t) = \psi(t)\epsilon(t)/t$. Moreover, the three conditions of (V.74) are satisfied. Thus $\psi(t) \in RV(0)$ and $h(x)$ is a rapidly varying function.

Denote by $\mathcal{R}$ the class of functions with either regular variation defined as in Case 2.2 or with rapid variation defined as in Case 2.2.

We now state hypotheses pertaining to the bounded function q in (V.67). We assume that

$$|q(x)| \in RV(\eta), \text{ for some } \eta < \theta - \tfrac{3\beta}{2} - \tfrac{3}{2} \text{ if } h \in RV(\beta) \tag{V.18}$$

and

$$|q(\psi(t))| \in RV(\eta), \text{ for some } \eta < -\tfrac{1}{2} \text{ if } h \text{ is rapidly varying.} \tag{V.19}$$

1.3 An Abelian-type theorem

We have

Theorem 5. *Let $p(x)$ be defined as in (V.67) and $h(x)$ belong to $\mathcal{R}$. Denote by $m(t)$, $s^2(t)$ and*

$\mu_j(t)$ for $j = 3, 4, ...$ the functions defined in (V.6), (V.7) and (V.8). Then it holds

$$\begin{aligned}
m(t) &\underset{t\to\infty}{=} \psi(t)(1+o(1)),\\
s^2(t) &\underset{t\to\infty}{=} \psi'(t)(1+o(1)),\\
\mu_3(t) &\underset{t\to\infty}{=} \psi^{(2)}(t)(1+o(1)),\\
\mu_j(t) &\underset{t\to\infty}{=} \begin{cases} M_j s^j(t)(1+o(1)), & \text{for even } j>3\\ \frac{(M_{j+3}-3jM_{j-1})\mu_3(t)s^{j-3}(t)}{6}(1+o(1)), & \text{for odd } j>3 \end{cases},
\end{aligned}$$

where M_i, $i > 0$, denotes the ith order moment of standard normal distribution.

Using (V.67), the moment generating function $\Phi(t)$ defined in (V.4) takes on the form

$$\Phi(t) = \int_0^\infty e^{tx} p(x)dx = c\int_0^\infty \exp(K(x,t)+q(x))dx, \qquad t\in(0,\infty)$$

where

$$K(x,t) = tx - g(x). \tag{V.20}$$

If $h \in \mathcal{R}$, then for fixed t, $x \mapsto K(x,t)$ is a concave function and takes its maximum value at $\hat{x} = h^{\leftarrow}(t)$.

As a direct by-product of Th. 8 we obtain the following Abel type result.

Theorem 6. *Under the same hypotheses as in Th. 8, we have*

$$\Phi(t) = \sqrt{2\pi}\hat{\sigma}e^{K(\hat{x},t)}(1+o(1)).$$

Remark 3. *It is easily verified that this result is in accordance with Th. 4.12.11 of (Bingham et al., 1987), Th. 3 of (Borovkov, 2008) and Th. 4.2 of (Juszczak and Nagaev, 2004). Some classical consequence of Kasahara's Tauberian theorem can be paralleled with Th. 6. Following Th. 4.2.10 in (Bingham et al., 1987), with f defined as g above, it follows that $-\log\int_x^\infty p(v)dv \sim g(x)$ as $x\to\infty$ under Case 2.2, a stronger assumption than required in Th. 4.2.10 of (Bingham et al., 1987). Th. 4.12.7 in (Bingham et al., 1987) hence applies and provides an asymptotic equivalent for $\log\Phi(t)$ as $t\to\infty$; Th. 6 improves on this result, at the cost of the additional regularity assumptions of Case 2.2. Furthermore, these results complement those in (Broniatowski and Celant, 2014) Sect. 3.2, in Case 2.2.*

We also derive the following consequence of Th. 8.

Theorem 7. *Under the present hypotheses, denote $\mathfrak{X}_t$ a random variable with density $\pi_t(x)$. Then as $t \to \infty$, the family of random variables*

$$\frac{\mathfrak{X}_t - m(t)}{s(t)}$$

converges in distribution to a standard normal distribution.

Remark 4. *This result holds under various hypotheses, as developped for example in (Balkema et al., 1993) or (Feigin and Yashchin, 1983). Under log-concavity of p it also holds locally; namely the family of densities π_t converges pointwise to the standard gaussian density; this yields asymptotic results for the extreme deviations of the empirical mean of i.i.d. summands with light tails (see (Broniatowsk and Mason, 1994)), and also provides sufficient conditions for P_X to belong to the domain of attraction of the Gumbel distribution for the maximum, through criterions pertaining to the Mill's ratio (see (Feigin and Yashchin, 1983)).*

Remark 5. *That g is four times derivable can be relaxed; in Case 2.2 with $\beta > 2$ or in Case 2.2, g a three times derivable function, together with the two first lines in (V.71) and (V.74), provides Th. 8, 6 and 7. Also it may be seen that the order of differentiability of g in Case 2.2 with $0 < \beta \leq 2$ is related to the order of the moment of the tilted distribution for which an asymptotic equivalent is obtained. This will be developed in a forthcoming paper.*

The proofs of the above results rely on five lemmas exposed in Appendix.

1.4 Appendix

The proofs of the Abelian theorems rely on Lemmas 5 to 9. Lemma 5 is instrumental for Lemma 9.

The following lemma provides a simple argument for the local uniform convergence of regularly varying functions.

Lemma 1. *Consider $l(t) \in RV(\alpha), \alpha \in \mathbb{R}$. For any function f such that $f(t) \underset{t\to\infty}{=} o(t)$, it holds*

$$\sup_{|x|\leq f(t)} |l(t+x)| \underset{t\to\infty}{\sim} |l(t)|. \tag{V.21}$$

If $f(t) = at$ with $0 < a < 1$, then it holds

$$\sup_{|x|\leq at} |l(t+x)| \underset{t\to\infty}{\sim} (1+a)^{\alpha}|l(t)|. \tag{V.22}$$

Démonstration. By Th. 1.5.2 of (Bingham et al., 1987), if $l(t) \in RV(\alpha)$, then for all I

$$\sup_{\lambda \in I} \left| \frac{l(\lambda t)}{l(t)} - \lambda^{\alpha} \right| \underset{t \to \infty}{\longrightarrow} 0,$$

with $I = [A, B] (0 < A \leq B < \infty)$ if $\alpha = 0$, $I = (0, B] \, (0 < B < \infty)$ if $\alpha > 0$ and $I = [A, \infty) \, (0 < A < \infty)$ if $\alpha < 0$.

Putting $\lambda = 1 + x/t$ with $f(t) \underset{t \to \infty}{=} o(t)$, we obtain

$$\sup_{|x| \leq f(t)} \left| \frac{l(t+x)}{l(t)} \right| - \left(1 + \frac{f(t)}{t} \right)^{\alpha} \underset{t \to \infty}{\longrightarrow} 0,$$

which implies (V.21).

When $f(t) = at$ with $0 < a < 1$, we get

$$\sup_{|x| \leq at} \left| \frac{l(t+x)}{l(t)} \right| - (1+a)^{\alpha} \underset{t \to \infty}{\longrightarrow} 0,$$

which implies (V.22).

■

Now we quote some simple expansions pertaining to the function h under the two cases considered in the above section.

Lemma 2. *We have under Case 2.2,*

$$\begin{aligned} h'(x) &= \frac{h(x)}{x}[\beta + \epsilon(x)], \\ h^{(2)}(x) &= \frac{h(x)}{x^2}[\beta(\beta-1) + a\epsilon(x) + \epsilon^2(x) + x\epsilon'(x)], \\ h^{(3)}(x) &= \frac{h(x)}{x^3}[\beta(\beta-1)(\beta-2) + b\epsilon(x) + c\epsilon^2(x) + \epsilon^3(x) \\ &\qquad + x\epsilon'(x)(d + e\epsilon(x)) + x^2\epsilon^{(2)}(x)]. \end{aligned}$$

where a, b, c, d, e are some real constants.

Corollary 1. *We have under Case 2.2, $h'(x) \underset{x \to \infty}{\sim} \beta h(x)/x$ and $|h^{(i)}(x)| \leq C_i h(x)/x^i, i = 1, 2, 3,$ for some constants C_i and for large x .*

Corollary 2. *We have under Case 2.2, $\hat{x}(t) = \psi(t) \in RV(1/\beta)$ (see Th. (1.5.15) of (Bingham et al., 1987)) and $\hat{\sigma}^2(t) = \psi'(t) \sim \beta^{-1}\psi(t)/t \in RV(1/\beta - 1)$.*

It also holds

Lemma 3. *We have under Case 2.2,*

$$\psi^{(2)}(t) \underset{t\to\infty}{\sim} -\frac{\psi(t)\epsilon(t)}{t^2} \text{ and } \psi^{(3)}(t) \underset{t\to\infty}{\sim} 2\frac{\psi(t)\epsilon(t)}{t^3}.$$

Lemma 4. *We have under Case 2.2,*

$$\begin{aligned} h'(\psi(t)) &= \frac{1}{\psi'(t)} = \frac{t}{\psi(t)\epsilon(t)}, \\ h^{(2)}(\psi(t)) &= -\frac{\psi^{(2)}(t)}{(\psi'(t))^3} \underset{t\to\infty}{\sim} \frac{t}{\psi^2(t)\epsilon^2(t)}, \\ h^{(3)}(\psi(t)) &= \frac{3(\psi^{(2)}(t))^2 - \psi^{(3)}(t)\psi'(t)}{(\psi'(t))^5} \underset{t\to\infty}{\sim} \frac{t}{\psi^3(t)\epsilon^3(t)}. \end{aligned}$$

Corollary 3. *We have under Case 2.2, $\hat{x}(t) = \psi(t) \in RV(0)$ and $\hat{\sigma}^2(t) = \psi'(t) = \psi(t)\epsilon(t)/t \in RV(-1)$. Moreover, we have $h^{(i)}(\psi(t)) \in RV(1), i = 1, 2, 3$.*

Before beginning the proofs of our results we quote that the regularity conditions (V.70) and (V.73) pertaining to the function h allow for the above simple expansions. Substituting the constant c in (V.70) and (V.73) by functions $x \to c(x)$ which converge smoothly to some positive constant c adds noticeable complexity.

We now come to the proofs of five lemmas which provide the asymptotics leading to Th. 8 and Th. 6.

Lemma 5. *It holds*

$$\frac{\log \hat{\sigma}}{\int_1^t \psi(u)du} \underset{t\to\infty}{\longrightarrow} 0.$$

Démonstration. By Cor. 2 and Cor. 3, we have that $\psi(t) \in RV(1/\beta)$ in Case 2.2 and $\psi(t) \in RV(0)$ in Case 2.2. Using Th. 1 of (Feller, 1971), Chap. 8.9 or Prop. 1.5.8 of (Bingham et al., 1987), we obtain

$$\int_1^t \psi(u)du \underset{t\to\infty}{\sim} \begin{cases} t\psi(t)/(1+1/\beta) \in RV(1+1/\beta) & \text{if } h \in RV(\beta) \\ t\psi(t) \in RV(1) & \text{if } h \text{ is rapidly varying} \end{cases}. \tag{V.23}$$

Also by Cor. 2 and 3, we have that $\hat{\sigma}^2 \in RV(1/\beta - 1)$ in Case 2.2 and $\hat{\sigma}^2 \in RV(-1)$ in Case 2.2. Thus $t \mapsto \log \hat{\sigma} \in RV(0)$ by composition and

$$\frac{\log \hat{\sigma}}{\int_1^t \psi(u)du} \underset{t\to\infty}{\sim} \begin{cases} \frac{\beta+1}{\beta} \times \frac{\log \hat{\sigma}}{t\psi(t)} \in RV\left(-1-\frac{1}{\beta}\right) & \text{if } h \in RV(\beta) \\ \frac{\log \hat{\sigma}}{t\psi(t)} \in RV(-1) & \text{if } h \text{ is rapidly varying} \end{cases},$$

which proves the claim. ■

The next steps of the proof make use of the function

$$L(t) := (\log t)^3 .$$

Lemma 6. *We have*

$$\sup_{|x|\le \hat{\sigma}L(t)} \left| \frac{h^{(3)}(\hat{x}+x)}{h^{(2)}(\hat{x})} \right| \hat{\sigma} L^4(t) \underset{t\to\infty}{\longrightarrow} 0.$$

Démonstration.

***Case* 1.** *By Cor. 1 and by (V.72) we have*

$$|h^{(3)}(x)| \le C \frac{|h^{(2)}(x)|}{x},$$

for some constant C and x large. Since, by Cor. 2, $\hat{x} \in RV(1/\beta)$ and $\hat{\sigma}^2 \in RV(1/\beta - 1)$, we have

$$\frac{|x|}{\hat{x}} \le \frac{\hat{\sigma}L(t)}{\hat{x}} \in RV\left(-\frac{1}{2} - \frac{1}{2\beta}\right)$$

and $|x|/\hat{x} \underset{t\to\infty}{\longrightarrow} 0$ uniformly in $\{x : |x| \le \hat{\sigma}L(t)\}$. For large t and all x such that $|x| \le \hat{\sigma}L(t)$, we have

$$|h^{(3)}(\hat{x}+x)| \le C \frac{|h^{(2)}(\hat{x}+x)|}{\hat{x}+x} \le C \sup_{|x|\le\hat{\sigma}L(t)} \frac{|h^{(2)}(\hat{x}+x)|}{\hat{x}+x}$$

whence

$$\sup_{|x|\le\hat{\sigma}L(t)} |h^{(3)}(\hat{x}+x)| \le C \sup_{|x|\le\hat{\sigma}L(t)} \frac{|h^{(2)}(\hat{x}+x)|}{\hat{x}+x}$$

where

$$\sup_{|x|\le\hat{\sigma}L(t)} \frac{|h^{(2)}(\hat{x}+x)|}{\hat{x}+x} \underset{t\to\infty}{\sim} \frac{|h^{(2)}(\hat{x})|}{\hat{x}},$$

using (V.21) for the regularly varying function $|h^{(2)}(\hat{x})| \in RV(\theta/\beta)$, *with* $f(t) = \hat{\sigma}L(t) \underset{t\to\infty}{=} o(\hat{x})$. *Thus for* t *large enough and for all* $\delta > 0$

$$\sup_{|x|\le\hat{\sigma}L^4(t)} \left|\frac{h^{(3)}(\hat{x}+x)}{h^{(2)}(\hat{x})}\right| \hat{\sigma}L^4(t) \le C\frac{\hat{\sigma}L^4(t)}{\hat{x}}(1+\delta) \in RV\left(\frac{1}{2\beta} - \frac{1}{2} - \frac{1}{\beta}\right),$$

which proves Lem. 6 in Case 2.2.

***Case* 2.** *By Lem. 4, we have that* $h^{(3)}(\psi(t)) \in RV(1)$. *Using V.22, we have for* $0 < a < 1$ *and* t *large enough*

$$\sup_{|v|\le at} |h^{(3)}(\psi(t+v))| \underset{t\to\infty}{\sim} (1+a)h^{(3)}(\psi(t)).$$

In the present case $\hat{x} \in RV(0)$ *and* $\hat{\sigma}^2 \in RV(-1)$. *Setting* $\psi(t+v) = \hat{x}+x = \psi(t)+x$, *we have* $x = \psi(t+v) - \psi(t)$ *and* $A := \psi(t-at) - \psi(t) \le x \le \psi(t+at) - \psi(t) =: B$, *since* $t \mapsto \psi(t)$ *is an increasing function. It follows that*

$$\sup_{|v|\le at} h^{(3)}(\psi(t+v)) = \sup_{A\le x\le B} h^{(3)}(\hat{x}+x).$$

Now note that (cf. page 127 in (Bingham et al., 1987))

$$B = \psi(t+at) - \psi(t) = \int_t^{t+at} \psi'(z)dz = \int_t^{t+at} \frac{\psi(z)\epsilon(z)}{z}dz \underset{t\to\infty}{\sim} \psi(t)\epsilon(t)\log(1+a),$$

since $\psi(t)\epsilon(t) \in RV(0)$. *Moreover, we have*

$$\frac{\hat{\sigma}L(t)}{\psi(t)\epsilon(t)} \in RV(-1) \text{ and } \frac{\hat{\sigma}L(t)}{\psi(t)\epsilon(t)} \underset{t\to\infty}{\longrightarrow} 0.$$

It follows that $\hat{\sigma}L(t) \underset{t\to\infty}{=} o(B)$ *and in a similar way, we have* $\hat{\sigma}L(t) \underset{t\to\infty}{=} o(A)$. *Using Lem. 4 and since* $\hat{\sigma}L^4(t) \in RV(-1/2)$, *it follows that for* t *large enough and for all* $\delta > 0$

$$\begin{aligned}\sup_{|x|\le\hat{\sigma}L(t)} \frac{|h^{(3)}(\psi(t+v))|}{|h^{(2)}(\psi(t))|}\hat{\sigma}L^4(t) &\le \sup_{A\le x\le B} \frac{|h^{(3)}(\psi(t+v))|}{|h^{(2)}(\psi(t))|}\hat{\sigma}L^4(t) \\ &\le (1+a)\frac{\hat{\sigma}L^4(t)}{\psi(t)\epsilon(t)}(1+\delta) \in RV\left(-\frac{1}{2}\right),\end{aligned}$$

which concludes the proof of Lem. 6 in Case 2.2.

■

Lemma 7. *We have*

$$|h^{(2)}(\hat{x})|\hat{\sigma}^4 \underset{t\to\infty}{\longrightarrow} 0,$$
$$|h^{(2)}(\hat{x})|\hat{\sigma}^3 L(t) \underset{t\to\infty}{\longrightarrow} 0.$$

Démonstration.

***Case* 1.** *By Cor. 1 and Cor. 2, we have*

$$|h^{(2)}(\hat{x})|\hat{\sigma}^4 \leq \frac{C_2}{\beta^2 t} \in RV(-1)$$

and

$$|h^{(2)}(\hat{x})|\hat{\sigma}^3 L(t) \leq \frac{C_2}{\beta^{3/2}} \frac{L(t)}{\sqrt{t\psi(t)}} \in RV\left(-\frac{1}{2\beta} - \frac{1}{2}\right),$$

proving the claim.

***Case* 2.** *We have by Lem. 4 and Cor. 3*

$$h^{(2)}(\hat{x})\hat{\sigma}^4 \underset{t\to\infty}{\sim} \frac{1}{t} \in RV(-1)$$

and

$$h^{(2)}(\hat{x})\hat{\sigma}^3 L(t) \underset{t\to\infty}{\sim} \frac{L(t)}{\sqrt{t\psi(t)\epsilon(t)}} \in RV\left(-\frac{1}{2}\right),$$

which concludes the proof of Lem. 7.

■

We now define some functions to be used in the sequel. A Taylor-Lagrange expansion of $K(x,t)$ in a neighborhood of $\hat{x}$ yields

$$K(x,t) = K(\hat{x},t) - \frac{1}{2}h'(\hat{x})(x-\hat{x})^2 - \frac{1}{6}h^{(2)}(\hat{x})(x-\hat{x})^3 + \varepsilon(x,t), \tag{V.24}$$

where, for some $\theta \in (0,1)$,

$$\varepsilon(x,t) = -\frac{1}{24}h^{(3)}(\hat{x} + \theta(x-\hat{x}))(x-\hat{x})^4. \tag{V.25}$$

Lemma 8. *We have*

$$\sup_{y\in[-L(t),L(t)]} \frac{|\xi(\hat\sigma y+\hat x,t)|}{h^{(2)}(\hat x)\hat\sigma^3} \underset{t\to\infty}{\longrightarrow} 0,$$

where $\xi(x,t)=\varepsilon(x,t)+q(x)$ *and* $\varepsilon(x,t)$ *is defined in (V.25).*

Démonstration. For $y\in[-L(t),L(t)]$, by V.25, it holds

$$\left|\frac{\varepsilon(\hat\sigma y+\hat x,t)}{h^{(2)}(\hat x)\hat\sigma^3}\right| \le \left|\frac{h^{(3)}(\hat x+\theta\hat\sigma y)(\hat\sigma y)^4}{h^{(2)}(\hat x)\hat\sigma^3}\right| \le \left|\frac{h^{(3)}(\hat x+\theta\hat\sigma y)\hat\sigma L^4(t)}{h^{(2)}(\hat x)}\right|,$$

with $\theta\in(0,1)$. Let $x=\theta\hat\sigma y$. It then holds $|x|\le\hat\sigma L(t)$. Therefore by Lem. 6

$$\sup_{y\in[-L(t),L(t)]}\left|\frac{\varepsilon(\hat\sigma y+\hat x,t)|}{h^{(2)}(\hat x)\hat\sigma^3}\right| \le \sup_{|x|\le\hat\sigma L(t)}\left|\frac{h^{(3)}(\hat x+x)}{h^{(2)}(\hat x)}\hat\sigma L^4(t)\right| \underset{t\to\infty}{\longrightarrow} 0.$$

It remains to prove that

$$\sup_{y\in[-L(t),L(t)]}\left|\frac{q(\hat\sigma y+\hat x)}{h^{(2)}(\hat x)\hat\sigma^3}\right| \underset{t\to\infty}{\longrightarrow} 0. \tag{V.26}$$

***Case* 1.** *By (V.72) and by composition,* $|h^{(2)}(\hat x)|\in RV(\theta/\beta)$. *Using Cor. 1 we obtain*

$$|h^{(2)}(\hat x)\hat\sigma^3| \underset{t\to\infty}{\sim} \frac{|h^{(2)}(\hat x)|\psi^{3/2}(t)}{\beta^{3/2}t^{3/2}} \in RV\left(\frac{\theta}{\beta}+\frac{3}{2\beta}-\frac{3}{2}\right).$$

Since, by (V.75), $|q(\hat x)|\in RV(\eta/\beta)$, *for* $\eta<\theta-3\beta/2+3/2$ *and putting* $x=\hat\sigma y$, *we obtain*

$$\sup_{y\in[-L(t),L(t)]}\left|\frac{q(\hat\sigma y+\hat x)}{h^{(2)}(\hat x)\hat\sigma^3}\right| = \sup_{|x|\le\hat\sigma L(t)}\left|\frac{q(\hat x+x)}{h^{(2)}(\hat x)\hat\sigma^3}\right| \underset{t\to\infty}{\sim} \frac{|q(\hat x)|}{|h^{(2)}(\hat x)\hat\sigma^3|} \in RV\left(\frac{\eta-\theta}{\beta}-\frac{3}{2\beta}+\frac{3}{2}\right),$$

which proves (V.26).

***Case* 2.** *By Lem. 4 and Cor. 3, we have*

$$|h^{(2)}(\hat x)\hat\sigma^3| \underset{t\to\infty}{\sim} \frac{1}{\sqrt{t\psi(t)\epsilon(t)}} \in RV\left(-\frac{1}{2}\right).$$

As in Lem. 6, since by (V.76), $q(\psi(t)) \in RV(\eta)$, then we obtain, with $\eta < -1/2$

$$\begin{aligned}\sup_{y\in[-L(t),L(t)]}\left|\frac{q(\hat{\sigma}y+\hat{x})}{h^{(2)}(\hat{x})\hat{\sigma}^3}\right| &= \sup_{|x|\le\hat{\sigma}L(t)}\left|\frac{q(\hat{x}+x)}{h^{(2)}(\hat{x})\hat{\sigma}^3}\right| \\ &\le \sup_{|v|\le at}\left|\frac{q(\psi(t+v))}{h^{(2)}(\hat{x})\hat{\sigma}^3}\right| \\ &\le (1+a)^{\eta}q(\psi(t))\sqrt{t\psi(t)\epsilon(t)}(1+\delta)\in RV\left(\eta+\frac{1}{2}\right),\end{aligned}$$

for all $\delta > 0$, with $a < 1$, t large enough and $\eta + 1/2 < 0$. This proves (V.26).

■

Lemma 9. *For $\alpha \in \mathbb{N}$, denote*

$$\Psi(t,\alpha) := \int_0^{\infty}(x-\hat{x})^{\alpha}e^{tx}p(x)dx.$$

Then

$$\Psi(t,\alpha) \underset{t\to\infty}{=} \hat{\sigma}^{\alpha+1}e^{K(\hat{x},t)}T_1(t,\alpha)(1+o(1)),$$

where

$$T_1(t,\alpha) = \int_{-\frac{L^{1/3}(t)}{\sqrt{2}}}^{\frac{L^{1/3}(t)}{\sqrt{2}}} y^{\alpha}\exp(-\frac{y^2}{2})dy - \frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{L^{1/3}(t)}{\sqrt{2}}}^{\frac{L^{1/3}(t)}{\sqrt{2}}} y^{3+\alpha}\exp(-\frac{y^2}{2})dy. \tag{V.27}$$

Démonstration. We define the interval I_t as follows

$$I_t := \left(-\frac{L^{\frac{1}{3}}(t)\hat{\sigma}}{\sqrt{2}}, \frac{L^{\frac{1}{3}}(t)\hat{\sigma}}{\sqrt{2}}\right).$$

For large enough τ, when $t \to \infty$ we can partition$\mathbb{R}_+$ into

$$\mathbb{R}_+ = \{x : 0 < x < \tau\} \cup \{x : x \in \hat{x} + I_t\} \cup \{x : x \ge \tau, x \notin \hat{x} + I_t\},$$

where for $x > \tau$, $q(x) < \log 2$. Thus we have

$$p(x) < 2e^{-g(x)}. \tag{V.28}$$

For fixed τ, $\{x : 0 < x < \tau\} \cap \{x : x \in \hat{x} + I_t\} = \emptyset$. Therefore $\tau < \hat{x} - \frac{L^{\frac{1}{3}}(t)\hat{\sigma}}{\sqrt{2}} \leq \hat{x}$ for t large enough. Hence it holds

$$\Psi(t,\alpha) =: \Psi_1(t,\alpha) + \Psi_2(t,\alpha) + \Psi_3(t,\alpha), \tag{V.29}$$

where

$$\begin{aligned}
\Psi_1(t,\alpha) &= \int_0^\tau (x-\hat{x})^\alpha e^{tx} p(x)dx, \\
\Psi_2(t,\alpha) &= \int_{x\in\hat{x}+I_t} (x-\hat{x})^\alpha e^{tx} p(x)dx, \\
\Psi_3(t,\alpha) &= \int_{x\notin\hat{x}+I_t, x\geq\tau} (x-\hat{x})^\alpha e^{tx} p(x)dx.
\end{aligned}$$

We estimate $\Psi_1(t,\alpha)$, $\Psi_2(t,\alpha)$ and $\Psi_3(t,\alpha)$ in Step 1, Step 2 and Step 3.

Step 1 : Since q is bounded, we consider

$$\log d = \sup_{x\in(0,\tau)} q(x)$$

and for t large enough, we have

$$|\Psi_1(t,\alpha)| \leq \int_0^\tau |x-\hat{x}|^\alpha e^{tx} p(x)dx \leq d\int_0^\tau \hat{x}^\alpha e^{tx}dx,$$

since when $0 < x < \tau < \hat{x}$ then $|x-\hat{x}| = \hat{x} - x < \hat{x}$ for t large enough and g is positive. Since, for t large enough, we have

$$\int_0^\tau \hat{x}^\alpha e^{tx}dx = \hat{x}^\alpha \frac{e^{t\tau}}{t} - \frac{\hat{x}^\alpha}{t} \leq \hat{x}^\alpha \frac{e^{t\tau}}{t},$$

we obtain

$$|\Psi_1(t,\alpha)| \leq d\hat{x}^\alpha \frac{e^{t\tau}}{t}. \tag{V.30}$$

We now show that for $h \in \mathcal{R}$, it holds

$$\hat{x}^\alpha \frac{e^{t\tau}}{t} \underset{t\to\infty}{=} o(|\hat{\sigma}^{\alpha+1}| e^{K(\hat{x},t)} |h^{(2)}(\hat{x})\hat{\sigma}^3|), \tag{V.31}$$

with $K(x,t)$ defined as in (V.20). This is equivalent to

$$\frac{\hat{x}^\alpha e^{t\tau}}{t|\hat{\sigma}^{\alpha+4}h^{(2)}(\hat{x})|} \underset{t\to\infty}{=} o(e^{K(\hat{x},t)}),$$

which is implied by

$$-(\alpha+4)\log|\hat{\sigma}| - \log t + \alpha\log\hat{x} + \tau t - \log|h^{(2)}(\hat{x})| \underset{t\to\infty}{=} o(K(\hat{x},t)), \tag{V.32}$$

if $K(\hat{x},t) \underset{t\to\infty}{\longrightarrow} \infty$.

Setting $u = h(v)$ in $\int_1^t \psi(u)du$, we have

$$\int_1^t \psi(u)du = t\psi(t) - \psi(1) - g(\psi(t)) + g(\psi(1)).$$

Since $K(\hat{x},t) = t\psi(t) - g(\psi(t))$, we obtain

$$K(\hat{x},t) = \int_1^t \psi(u)du + \psi(1) - g(\psi(1)) \underset{t\to\infty}{\sim} \int_1^t \psi(u)du. \tag{V.33}$$

Let us denote (V.23) by

$$K(\hat{x},t) \underset{t\to\infty}{\sim} at\psi(t), \tag{V.34}$$

with

$$a = \begin{cases} \frac{\beta}{\beta+1} & \text{if } h \in RV(\beta) \\ 1 & \text{if } h \text{ is rapidly varying} \end{cases}.$$

We have to show that each term in (V.32) is $o(K(\hat{x},t))$.

1. By Lem. 5, $\log\hat{\sigma} \underset{t\to\infty}{=} o(\int_1^t \psi(u)du)$. Hence $\log\hat{\sigma} \underset{t\to\infty}{=} o(K(\hat{x},t))$.
2. By Cor. 2 and Cor. 3, we have

$$\frac{t}{K(\hat{x},t)} \underset{t\to\infty}{\sim} \frac{1}{a\psi(t)} \underset{t\to\infty}{\longrightarrow} 0.$$

   Thus $t \underset{t\to\infty}{=} o(K(\hat{x},t))$.
3. Since $\hat{x} = \psi(t) \underset{t\to\infty}{\longrightarrow} \infty$, it holds

$$\left|\frac{\log\hat{x}}{K(\hat{x},t)}\right| \le C\frac{\psi(t)}{K(\hat{x},t)},$$

for some positive constant C and t large enough. Moreover by (V.34), we have

$$\frac{\psi(t)}{K(\hat{x},t)} \underset{t\to\infty}{\sim} \frac{1}{at} \underset{t\to\infty}{\longrightarrow} 0.$$

Hence $\log \hat{x} \underset{t\to\infty}{=} o(K(\hat{x},t))$.

4. Using (V.34), $\log|h^{(2)}(\hat{x})| \in RV(0)$ and $\log|h^{(2)}(\hat{x})| \underset{t\to\infty}{=} o(K(\hat{x},t))$.
5. Since $\log t \underset{t\to\infty}{=} o(t)$ and $t \underset{t\to\infty}{=} o(K(\hat{x},t))$, we obtain $\log t \underset{t\to\infty}{=} o(K(\hat{x},t))$.

Since (V.32) holds and $K(\hat{x},t) \underset{t\to\infty}{\longrightarrow} \infty$ by (V.33) and (V.34), we then get (V.31).

(V.30) and (V.31) yield together

$$|\Psi_1(t,\alpha)| \underset{t\to\infty}{=} o(|\hat{\sigma}^{\alpha+1}|e^{K(\hat{x},t)}|h^{(2)}(\hat{x})\hat{\sigma}^3|). \tag{V.35}$$

When α is even,

$$T_1(t,\alpha) = \int_{-\frac{t^{1/3}}{\sqrt{2}}}^{\frac{t^{1/3}}{\sqrt{2}}} y^\alpha \exp(-\frac{y^2}{2})dy \underset{t\to\infty}{\sim} \sqrt{2\pi}M_\alpha, \tag{V.36}$$

where M_α is the moment of order α of a standard normal distribution. Thus by Lem. 7 we have

$$\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{T_1(t,\alpha)} \underset{t\to\infty}{\longrightarrow} 0. \tag{V.37}$$

When α is odd,

$$T_1(t,\alpha) = -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{t^{\frac{1}{3}}}{\sqrt{2}}}^{\frac{t^{\frac{1}{3}}}{\sqrt{2}}} y^{3+\alpha}\exp\left(-\frac{y^2}{2}\right)dy \underset{t\to\infty}{\sim} -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\sqrt{2\pi}M_{\alpha+3}, \tag{V.38}$$

where $M_{\alpha+3}$ is the moment of order $\alpha+3$ of a standard normal distribution. Thus we have

$$\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{T_1(t,\alpha)} \underset{t\to\infty}{\sim} -\frac{6}{\sqrt{2\pi}M_{\alpha+3}}. \tag{V.39}$$

Combined with (V.35), (V.37) and (V.39) imply for $\alpha \in \mathbb{N}$

$$|\Psi_1(t,\alpha)| \underset{t\to\infty}{=} o(\hat{\sigma}^{\alpha+1}e^{K(\hat{x},t)}T_1(t,\alpha)). \tag{V.40}$$

Step 2 : By (V.67) and (V.24)

$$\begin{aligned}\Psi_2(t,\alpha) &= \int_{x\in\hat{x}+I_t}(x-\hat{x})^\alpha e^{K(x,t)+q(x)}dx \\ &= \int_{x\in\hat{x}+I_t}(x-\hat{x})^\alpha e^{K(\hat{x},t)-\frac{1}{2}h'(\hat{x})(x-\hat{x})^2-\frac{1}{6}h^{(2)}(\hat{x})(x-\hat{x})^3+\xi(x,t)}dx,\end{aligned}$$

where $\xi(x,t)=\varepsilon(x,t)+q(x)$. Making the substitution $y=(x-\hat{x})/\hat{\sigma}$, it holds

$$\Psi_2(t,\alpha)=\hat{\sigma}^{\alpha+1}e^{K(\hat{x},t)}\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^\alpha \exp\left(-\frac{y^2}{2}-\frac{\hat{\sigma}^3y^3}{6}h^{(2)}(\hat{x})+\xi(\hat{\sigma}y+\hat{x},t)\right)dy, \tag{V.41}$$

since $h'(\hat{x})=1/\hat{\sigma}^2$.

On $\left\{y : y\in\left(-L^{\frac{1}{3}}(t)/\sqrt{2}, L^{\frac{1}{3}}(t)/\sqrt{2}\right)\right\}$, by Lem. 7, we have

$$\left|h^{(2)}(\hat{x})\hat{\sigma}^3y^3\right| \le \left|h^{(2)}(\hat{x})\hat{\sigma}^3L(t)\right|/2^{\frac{3}{2}} \underset{t\to\infty}{\longrightarrow} 0.$$

Perform the first order Taylor expansion

$$\exp\left(-\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}y^3+\xi(\hat{\sigma}y+\hat{x},t)\right) \underset{t\to\infty}{=} 1-\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}y^3+\xi(\hat{\sigma}y+\hat{x},t)+o_1(t,y),$$

where

$$o_1(t,y)=o\left(-\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}y^3+\xi(\hat{\sigma}y+\hat{x},t)\right). \tag{V.42}$$

We obtain

$$\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^\alpha \exp\left(-\frac{y^2}{2}-\frac{\hat{\sigma}^3y^3}{6}h^{(2)}(\hat{x})+\xi(\hat{\sigma}y+\hat{x},t)\right)dy =: T_1(t,\alpha)+T_2(t,\alpha),$$

where $T_1(t,\alpha)$ is defined in (V.27) and

$$T_2(t,\alpha) := \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} \left(\xi(\hat{\sigma}y+\hat{x},t)+o_1(t,y)\right)y^\alpha e^{-\frac{y^2}{2}}dy. \tag{V.43}$$

Using (V.42) we have for t large enough

$$|T_2(t,\alpha)| \leq \sup_{y\in[-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}},\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}]} |\xi(\hat{\sigma}y+\hat{x},t)| \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} |y|^{\alpha} e^{-\frac{y^2}{2}} dy$$
$$+ \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} \left(\left| o\left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6} y^3 \right) \right| + |o(\xi(\hat{\sigma}y+\hat{x},t))| \right) |y|^{\alpha} e^{-\frac{y^2}{2}} dy,$$

where $\sup_{y\in[-L^{\frac{1}{3}}(t)/\sqrt{2},L^{\frac{1}{3}}(t)/\sqrt{2}]} |\xi(\hat{\sigma}y+\hat{x},t)| \leq \sup_{y\in[-L(t),L(t)]} |\xi(\hat{\sigma}y+\hat{x},t)|$ since $L^{\frac{1}{3}}(t)/\sqrt{2} \leq L(t)$ holds for t large enough. Thus

$$|T_2(t,\alpha)| \leq 2 \sup_{y\in[-L(t),L(t)]} |\xi(\hat{\sigma}y+\hat{x},t)| \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} |y|^{\alpha} e^{-\frac{y^2}{2}} dy$$
$$+ \left| o\left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6} \right) \right| \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} |y|^{3+\alpha} e^{-\frac{y^2}{2}} dy$$
$$\underset{t\to\infty}{=} \left| o\left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6} \right) \right| \left(\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} |y|^{\alpha} e^{-\frac{y^2}{2}} dy + \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} |y|^{3+\alpha} e^{-\frac{y^2}{2}} dy \right),$$

where the last equality holds from Lem. 8. Since the integrals in the last equality are both bounded, it holds

$$T_2(t,\alpha) \underset{t\to\infty}{=} o(h^{(2)}(\hat{x})\hat{\sigma}^3). \tag{V.44}$$

When α is even, using (V.36) and Lem. 7

$$\left| \frac{T_2(t,\alpha)}{T_1(t,\alpha)} \right| \leq \frac{|h^{(2)}(\hat{x})\hat{\sigma}^3|}{\sqrt{2\pi} M_{\alpha}} \underset{t\to\infty}{\longrightarrow} 0. \tag{V.45}$$

When α is odd, using (V.38), we get

$$\frac{T_2(t,\alpha)}{T_1(t,\alpha)} \underset{t\to\infty}{=} -\frac{6}{\sqrt{2\pi} M_{\alpha+3}} o(1) \underset{t\to\infty}{\longrightarrow} 0. \tag{V.46}$$

Now with $\alpha \in \mathbb{N}$, by (V.45) and (V.46)

$$T_2(t,\alpha) \underset{t\to\infty}{=} o(T_1(t,\alpha)),$$

which, combined with (V.41), yields

$$\Psi_2(t,\alpha) = c\hat{\sigma}^{\alpha+1} e^{K(\hat{x},t)} T_1(t,\alpha)(1+o(1)). \tag{V.47}$$

Step 3 : The Three Chords Lemma implies, for $x \mapsto K(x,t)$ concave and $(x,y,z) \in \mathbb{R}_+^3$ such that $x < y < z$

$$\frac{K(y,t)-K(z,t)}{y-z} \leq \frac{K(x,t)-K(z,t)}{x-z} \leq \frac{K(x,t)-K(y,t)}{x-y}. \tag{V.48}$$

Since $x \mapsto K(x,t)$ is concave and attains its maximum in $\hat{x}$, we consider two cases : $x < \hat{x}$ and $x \geq \hat{x}$. After some calculus using (V.48) in each case, we get

$$K(x,t) - K(\hat{x},t) \leq \frac{K(\hat{x} + sgn(x-\hat{x})\frac{L^{1/3}(t)\hat{\sigma}}{\sqrt{2}}) - K(\hat{x},t)}{sgn(x-\hat{x})\frac{L^{1/3}(t)\hat{\sigma}}{\sqrt{2}}}(x-\hat{x}), \tag{V.49}$$

where

$$sgn(x-\hat{x}) = \begin{cases} 1 & \text{if } x \geq \hat{x} \\ -1 & \text{if } x < \hat{x} \end{cases}.$$

Using Lem. 7, a third-order Taylor expansion in the numerator of (V.49) gives

$$K(\hat{x} + sgn(x-\hat{x})\frac{L^{1/3}(t)\hat{\sigma}}{\sqrt{2}}) - K(\hat{x},t) \leq -\frac{1}{4}h'(\hat{x})L^{2/3}(t)\hat{\sigma}^2 = -\frac{1}{4}L^{2/3}(t),$$

which yields

$$K(x,t) - K(\hat{x},t) \leq -\frac{\sqrt{2}}{4}\frac{L^{1/3}(t)}{\hat{\sigma}}|x-\hat{x}|.$$

Using (V.28), we obtain for large enough fixed τ

$$\begin{aligned}
|\Psi_3(t,\alpha)| &\leq 2\int_{x \notin \hat{x}+I_t, x>\tau} |x-\hat{x}|^\alpha e^{K(x,t)} dx \\
&\leq 2e^{K(\hat{x},t)} \int_{|x-\hat{x}|>\frac{L^{1/3}(t)\hat{\sigma}}{\sqrt{2}}, x>\tau} |x-\hat{x}|^\alpha \exp\left(-\frac{\sqrt{2}}{4}\frac{L^{1/3}(t)}{\hat{\sigma}}|x-\hat{x}|\right) dx \\
&= 2e^{K(\hat{x},t)}\hat{\sigma}^{\alpha+1}\left[\int_{\frac{L^{1/3}(t)}{\sqrt{2}}}^{+\infty} y^\alpha e^{-\frac{\sqrt{2}}{4}L^{1/3}(t)y} dy + \int_{\frac{\tau-\hat{x}}{\hat{\sigma}}}^{-\frac{L^{1/3}(t)}{\sqrt{2}}} (-y)^\alpha e^{\frac{\sqrt{2}}{4}L^{1/3}(t)y} dy\right] \\
&:= 2e^{K(\hat{x},t)}\hat{\sigma}^{\alpha+1}(I_\alpha + J_\alpha).
\end{aligned}$$

It is easy but a bit tedious to show by recursion that

$$\begin{aligned} I_\alpha &= \int_{\frac{L^{1/3}(t)}{\sqrt{2}}}^{+\infty} y^\alpha \exp\left(-\frac{\sqrt{2}}{4}L^{1/3}(t)y\right) dy \\ &= \exp(-\frac{1}{4}L^{2/3}(t)) \sum_{i=0}^{\alpha} 2^{\frac{4i+3-\alpha}{2}} L^{\frac{\alpha-(2i+1)}{3}}(t) \frac{\alpha!}{(\alpha-i)!} \\ &\underset{t\to\infty}{\sim} 2^{\frac{3-\alpha}{2}} \exp(-\frac{1}{4}L^{2/3}(t)) L^{\frac{\alpha-1}{3}}(t) \end{aligned}$$

and

$$\begin{aligned} J_\alpha &= \int_{\frac{\tau-\hat{x}}{\hat{\sigma}}}^{-\frac{L^{1/3}(t)}{\sqrt{2}}} (-y)^\alpha \exp\left(\frac{\sqrt{2}}{4}L^{1/3}(t)y\right) dy \\ &= I_\alpha - \exp\left(\frac{\sqrt{2}}{4}L^{1/3}(t)\frac{\tau-\hat{x}}{\hat{\sigma}}\right) \sum_{i=0}^{\alpha} \left(\frac{\hat{x}-\tau}{\hat{\sigma}}\right)^{\alpha-i} L^{-\frac{i+1}{3}}(t) 2^{\frac{3i+3}{2}} \frac{\alpha!}{(\alpha-i)!} \\ &= I_\alpha + M(t), \end{aligned}$$

with $\hat{x}/\hat{\sigma} \in RV((1+1/\beta)/2)$ when $h \in RV(\beta)$ and $\hat{x}/\hat{\sigma} \in RV(1/2)$ when h is rapidly varying. Moreover, $\tau - \hat{x} < 0$, thus $M(t) \underset{t\to\infty}{\longrightarrow} 0$ and we have for some positive constant Q_1

$$|\Psi_3(t,\alpha)| \le Q_1 e^{K(\hat{x},t)} \hat{\sigma}^{\alpha+1} \exp(-\frac{1}{4}L^{2/3}(t)) L^{\frac{\alpha-1}{3}}(t).$$

With (V.47), we obtain for some positive constant Q_2

$$\left|\frac{\Psi_3(t,\alpha)}{\Psi_2(t,\alpha)}\right| \le \frac{Q_2 \exp(-\frac{1}{4}L^{2/3}(t)) L^{\frac{\alpha-1}{3}}(t)}{|T_1(t,\alpha)|}.$$

In Step 1, we saw that $T_1(t,\alpha) \underset{t\to\infty}{\sim} \sqrt{2\pi} M_\alpha$, for α even and $T_1(t,\alpha) \underset{t\to\infty}{\sim} -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\sqrt{2\pi} M_{\alpha+3}$, for α odd. Hence for α even and t large enough

$$\left|\frac{\Psi_3(t,\alpha)}{\Psi_2(t,\alpha)}\right| \le Q_3 \frac{\exp(-\frac{1}{4}L^{2/3}(t)) L^{\frac{\alpha-1}{3}}(t)}{\sqrt{2\pi} M_\alpha} \underset{t\to\infty}{\longrightarrow} 0, \tag{V.50}$$

and for α odd and t large enough

$$\left|\frac{\Psi_3(t,\alpha)}{\Psi_2(t,\alpha)}\right| \le Q_4 \frac{\exp(-\frac{1}{4}L^{2/3}(t)) L^{\frac{\alpha-1}{3}}(t)}{\frac{|h^{(2)}(\hat{x})\hat{\sigma}^3|}{6}\sqrt{2\pi} M_{\alpha+3}},$$

for positive constants Q_3 and Q_4.

As in Lem. 7, we have

$$|h^{(2)}(\hat{x})\hat{\sigma}^3| \in RV\left(\frac{\theta}{\beta}+\frac{3}{2\beta}-\frac{3}{2}\right) \text{ if } h \in RV(\beta)$$

and

$$|h^{(2)}(\hat{x})\hat{\sigma}^3| \in RV\left(-\frac{1}{2}\right) \text{ if } h \text{ is rapidly varying.}$$

Let us denote

$$|h^{(2)}(\hat{x})\hat{\sigma}^3| = t^{\rho}L_1(t),$$

for some slowly varying function L_1 and $\rho < 0$ defined as

$$\rho = \begin{cases} \frac{\theta}{\beta}+\frac{3}{2\beta}-\frac{3}{2} & \text{if } h \in RV(\beta) \\ -\frac{1}{2} & \text{if } h \text{ is rapidly varying} \end{cases}.$$

We have for some positive constant C

$$\left|\frac{\Psi_3(t,\alpha)}{\Psi_2(t,\alpha)}\right| \leq C\exp\left(-\frac{1}{4}L^{2/3}(t)-\rho\log t-\log L_1(t)\right)L^{\frac{\alpha-1}{3}}(t) \underset{t\to\infty}{\longrightarrow} 0,$$

since $-(\log t)^2/4-\rho\log t-\log L_1(t) \underset{t\to\infty}{\sim} -(\log t)^2/4 \underset{t\to\infty}{\longrightarrow} -\infty$.

Hence we obtain

$$\Psi_3(t,\alpha) \underset{t\to\infty}{=} o(\Psi_2(t,\alpha)). \tag{V.51}$$

The proof is completed by combining (V.29), (V.40), (V.47) and (V.51). ■

Proof of Th. 8. By Lem. 9, if $\alpha = 0$, it holds

$$T_1(t,0) \underset{t\to\infty}{\longrightarrow} \sqrt{2\pi},$$

since $L(t) \underset{t\to\infty}{\longrightarrow} \infty$. Approximate the moment generating function of X

$$\Phi(t) = \Psi(t,0) \underset{t\to\infty}{=} \hat{\sigma}e^{K(\hat{x},t)}T_1(t,0)(1+o(1)) \underset{t\to\infty}{=} \sqrt{2\pi}\hat{\sigma}e^{K(\hat{x},t)}(1+o(1)). \tag{V.52}$$

If $\alpha = 1$, it holds

$$T_1(t,1) \underset{t\to\infty}{=} -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}M_4\sqrt{2\pi}(1+o(1)),$$

where $M_4 = 3$ denotes the fourth order moment of the standard normal distribution. Consequently, we obtain

$$\Psi(t,1) \underset{t\to\infty}{=} -\sqrt{2\pi}\hat{\sigma}^2 e^{K(\hat{x},t)}\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{2}(1+o(1)) \underset{t\to\infty}{=} -\Phi(t)\frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}(1+o(1)), \tag{V.53}$$

which, together with the definition of $\Psi(t,\alpha)$, yields

$$\int_0^\infty xe^{tx}p(x)dx = \Psi(t,1) + \hat{x}\Phi(t) \underset{t\to\infty}{=} \left(\hat{x} - \frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}(1+o(1))\right)\Phi(t).$$

Hence we get

$$m(t) = \frac{\int_0^\infty xe^{tx}p(x)dx}{\Phi(t)} = \hat{x} - \frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}(1+o(1)). \tag{V.54}$$

By Lem. 7, we obtain

$$m(t) \underset{t\to\infty}{\sim} \hat{x} = \psi(t). \tag{V.55}$$

If $\alpha = 2$, it follows

$$T_1(t,2) \underset{t\to\infty}{=} \sqrt{2\pi}(1+o(1)).$$

Thus we have

$$\Psi(t,2) \underset{t\to\infty}{=} \hat{\sigma}^2\Phi(t)(1+o(1)). \tag{V.56}$$

Using (V.53), (V.54) and (V.56), it follows

$$\begin{aligned}
&\int_0^\infty (x-m(t))^2 e^{tx}p(x)dx = \int_0^\infty (x-\hat{x}+\hat{x}-m(t))^2 e^{tx}p(x)dx \\
&= \Psi(t,2) + 2(\hat{x}-m(t))\Psi(t,1) + (\hat{x}-m(t))^2\Phi(t) \\
&\underset{t\to\infty}{=} \hat{\sigma}^2\Phi(t)(1+o(1)) - \hat{\sigma}^2\Phi(t)\frac{(h^{(2)}(\hat{x})\hat{\sigma}^3)^2}{4}(1+o(1)) \underset{t\to\infty}{=} \hat{\sigma}^2\Phi(t)(1+o(1)),
\end{aligned}$$

where the last equality holds since $|h^{(2)}(\hat{x})\hat{\sigma}^3| \underset{t\to\infty}{\longrightarrow} 0$ by Lem. 7.

Hence we obtain

$$s^2(t) = \frac{\int_0^\infty (x-m(t))^2 e^{tx}p(x)dx}{\Phi(t)} \underset{t\to\infty}{\sim} \hat{\sigma}^2 = \psi'(t). \tag{V.57}$$

If $\alpha = 3$, it holds

$$T_1(t,3) = -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^6 e^{-\frac{y^2}{2}}dy.$$

Thus we have

$$\Psi(t,3) = -\sqrt{2\pi}\hat{\sigma}^4 e^{K(\hat{x},t)}\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}\frac{1}{\sqrt{2\pi}}y^6 e^{-\frac{y^2}{2}}dy \tag{V.58}$$
$$\underset{t\to\infty}{=} -M_6\frac{h^{(2)}(\hat{x})\hat{\sigma}^6}{6}\Phi(t)(1+o(1)),$$

where $M_6 = 15$ denotes the sixth order moment of standard normal distribution. Using (V.53), (V.54), (V.56) and (V.58), we have

$$\begin{aligned}
&\int_0^\infty (x-m(t))^3 e^{tx}p(x)dx = \int_0^\infty (x-\hat{x}+\hat{x}-m(t))^3 e^{tx}p(x)dx\\
&= \Psi(t,3) + 3(\hat{x}-m(t))\Psi(t,2) + 3(\hat{x}-m(t))^2\Psi(t,1) + (\hat{x}-m(t))^3\Phi(t)\\
&\underset{t\to\infty}{=} -h^{(2)}(\hat{x})\hat{\sigma}^6\Phi(t)(1+o(1)) - h^{(2)}(\hat{x})\hat{\sigma}^6\Phi(t)\frac{(h^{(2)}(\hat{x})\hat{\sigma}^3)^2}{4}(1+o(1))\\
&\underset{t\to\infty}{=} -h^{(2)}(\hat{x})\hat{\sigma}^6\Phi(t)(1+o(1)),
\end{aligned}$$

where the last equality holds since $|h^{(2)}(\hat{x})\hat{\sigma}^3| \underset{t\to\infty}{\longrightarrow} 0$ by Lem. 7. Hence we get

$$\mu_3(t) = \frac{\int_0^\infty (x-m(t))^3 e^{tx}p(x)dx}{\Phi(t)} \underset{t\to\infty}{\sim} -h^{(2)}(\hat{x})\hat{\sigma}^6 = \frac{\psi^{(2)}(t)}{(\psi'(t))^3}(\psi'(t))^3 = \psi^{(2)}(t). \tag{V.59}$$

We now consider $\alpha = j > 3$ for even j. Using (V.54) and Lem. 9, we have

$$\int_0^\infty (x-m(t))^j e^{tx}p(x)dx = \int_0^\infty (x-\hat{x}+\hat{x}-m(t))^j e^{tx}p(x)dx \tag{V.60}$$
$$= \sum_{i=0}^{j}\binom{j}{i}\left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}\right)^i \hat{\sigma}^{j-i+1}e^{K(\hat{x},t)}T_1(t,j-i)(1+o(1)),$$

with

$$T_1(t,j-i) = \begin{cases} \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^{j-i}e^{-\frac{y^2}{2}}dy & \text{for even } i\\ -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^{3+j-i}e^{-\frac{y^2}{2}}dy & \text{for odd } i \end{cases}$$
$$\underset{t\to\infty}{=} \begin{cases} \sqrt{2\pi}M_{j-i}(1+o(1)) & \text{if } i \text{ is even}\\ -\sqrt{2\pi}\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}M_{3+j-i} & \text{if } i \text{ is odd} \end{cases}.$$

Using (V.52), we obtain

$$\begin{aligned}
&\int_0^\infty (x-m(t))^j e^{tx} p(x)dx \\
&\underset{t\to\infty}{=} \sum_{i=0}^{j} \binom{j}{i} \left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}\right)^i \Phi(t)\times \\
&\left[\hat{\sigma}^{j-i} M_{j-i}(1+o(1))\mathbb{I}_{even\ i} - \frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}\sigma^{j-i-1}\frac{M_{3+j-i}}{3}(1+o(1))\mathbb{I}_{odd\ i}\right] \\
&\underset{t\to\infty}{=} \sum_{k=0}^{j/2} \binom{j}{2k} \left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}\right)^{2k} \Phi(t)\hat{\sigma}^{j-2k} M_{j-2k}(1+o(1)) \\
&- \sum_{k=0}^{j/2-1} \binom{j}{2k+1} \left(\frac{h^{(2)}(\hat{x})\hat{\sigma}^4}{2}\right)^{2(k+1)} \Phi(t)\hat{\sigma}^{j-2k-2}\frac{M_{3+j-2k-1}}{3}(1+o(1)) \\
&\underset{t\to\infty}{\sim} \hat{\sigma}^j \Phi(t)\times \\
&\left(M_j + \sum_{k=1}^{j/2} \binom{j}{2k} (h^{(2)}(\hat{x})\hat{\sigma}^3)^{2k}\frac{M_{j-2k}}{2^{2k}} - \sum_{k=0}^{j/2-1} \binom{j}{2k+1}(h^{(2)}(\hat{x})\hat{\sigma}^3)^{2(k+1)}\frac{M_{3+j-2k-1}}{3\times 2^{2(k+1)}}\right) \\
&\underset{t\to\infty}{=} M_j\hat{\sigma}^j\Phi(t)(1+o(1)),
\end{aligned}$$

since $|h^{(2)}(\hat{x})\hat{\sigma}^3| \underset{t\to\infty}{\longrightarrow} 0$ by Lem. 7. Hence we get for even j

$$\mu_j(t) = \frac{\int_0^\infty (x-m(t))^j e^{tx}p(x)dx}{\Phi(t)} \underset{t\to\infty}{\sim} M_j\hat{\sigma}^j \underset{t\to\infty}{\sim} M_j s^j(t), \tag{V.61}$$

by (V.57).

To conclude, we consider $\alpha = j > 3$ for odd j. (V.60) holds true with

$$\begin{aligned}
T_1(t, j-i) &= \begin{cases} \int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^{j-i}e^{-\frac{y^2}{2}}dy & \text{for odd } i \\ -\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}\int_{-\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}}^{\frac{L^{\frac{1}{3}}(t)}{\sqrt{2}}} y^{3+j-i}e^{-\frac{y^2}{2}}dy & \text{for even } i \end{cases} \\
&\underset{t\to\infty}{=} \begin{cases} \sqrt{2\pi}M_{j-i}(1+o(1)) & \text{if } i \text{ is odd} \\ -\sqrt{2\pi}\frac{h^{(2)}(\hat{x})\hat{\sigma}^3}{6}M_{3+j-i} & \text{if } i \text{ is even} \end{cases}.
\end{aligned}$$

Thus, with the same tools as above, some calculus and making use of (V.61),

$$\int_0^\infty (x-m(t))^j e^{tx} p(x)dx \underset{t\to\infty}{=} \frac{M_{j+3}-3jM_{j-1}}{6} \times (-h^{(2)}(\hat{x})\hat{\sigma}^{j+3})\Phi(t).$$

Hence we get for odd j

$$\mu_j(t) = \frac{\int_0^\infty (x-m(t))^j e^{tx} p(x)dx}{\Phi(t)} \underset{t\to\infty}{\sim} \frac{M_{j+3}-3jM_{j-1}}{6} \times (-h^{(2)}(\hat{x})\hat{\sigma}^{j+3}) \tag{V.62}$$
$$\underset{t\to\infty}{\sim} \frac{M_{j+3}-3jM_{j-1}}{6}\mu_3(t)s^{j-3}(t),$$

by (V.57) and (V.59).

The proof is complete by considering (V.55), (V.57), (V.59), (V.61) and (V.62). ■

Proof of Th. 6. It is proved incidentally in (V.52). ■

Proof of Th. 7. Consider the moment generating function of the random variable

$$Y_t := \frac{\mathfrak{X}_t - m(t)}{s(t)}.$$

It holds for any λ

$$\log E\exp \lambda Y_t = -\lambda\frac{m(t)}{s(t)} + \log\frac{\Phi\left(t+\frac{\lambda}{s(t)}\right)}{\Phi(t)}$$
$$= \frac{\lambda^2}{2}\frac{s^2\left(t+\theta\frac{\lambda}{s(t)}\right)}{s^2(t)} = \frac{\lambda^2}{2}\frac{\psi'\left(t+\theta\frac{\lambda(1+o(1))}{\sqrt{\psi'(t)}}\right)}{\psi'(t)}(1+o(1))$$

as $t\to\infty$, for some $\theta\in(0,1)$ depending on t, where we used Th. 8. Now making use of Cor. 2 and 3 it follows that

$$\lim_{t\to\infty} \log E\exp\lambda Y_t = \frac{\lambda^2}{2},$$

which proves the claim. ■

2 A Gibbs Conditional theorem under extreme deviation

Abstract We explore some properties of the conditional distribution of an i.i.d. sample under large exceedances of its sum. Thresholds for the asymptotic independance of the summands are observed, in contrast with the classical case when the conditioning event is in the range of a large deviation. This paper is an extension to Broniatowski and Cao (2012). Tools include a new Edgeworth expansion adapted to specific triangular arrays where the rows are generated by tilted distribution with diverging parameters, together with some Abelian type results.

2.1 Introduction

Let $X_1^n := (X_1, \ldots, X_n)$ be n independent unbounded real valued random variables and $S_1^n := X_1 + \ldots + X_n$ denote their sum. The purpose of this paper is to explore the limit distribution of the generic variable X_1 conditioned on extreme deviations (ED) pertaining to S_1^n. By extreme deviation we mean that S_1^n/n is supposed to take values which are going to infinity as n increases. Obviously such events are of infinitesimal probability. Our interest in this question stems from a first result which assesses that under appropriate conditions, when the sequence a_n is such that

$$\lim_{n\to\infty} a_n = \infty$$

then there exists a sequence ε_n which tends to 0 as n tends to infinity such that

$$\lim_{n\to\infty} P\left(\cap_{i=1}^n \left(X_i \in (a_n - \varepsilon_n, a_n + \varepsilon_n)\right) \middle| S_1^n/n > a_n\right) = 1, \tag{V.63}$$

which is to say that when the empirical mean takes exceedingly large values, then all the summands share the same behaviour. This result obviously requires a number of hypotheses, which we simply quote as "light tails" type. We refer to Broniatowski and Cao (2012) for this result and the connection with earlier related works; that such most unusual cases may be considered is argumented in this latest paper, in relation with the Erdös-Réniy law of large numbers and the formation of high level aggregates in random sequences.

The above result is clearly to be put in relation with the so-called Gibbs conditional Principle which we recall briefly in its simplest form.

Consider the case when the sequence $a_n = a$ is constant with value larger than the expectation of X_1. Hence we consider the behaviour of the summands when $(S_1^n/n > a)$, under a large

deviation (LD) condition about the empirical mean. The asymptotic conditional distribution of X_1 given $(S_1^n/n > a)$ is the well known tilted distribution of P_X with parameter t associated to a. Let us introduce some notation to shed some light on this The hypotheses to be stated now together with notation are kept throughout the entire paper. Without loss of generality it is assumed that the generic r.v. X_1 takes only non negative values.

It will be assumed that P_X, which is the distribution of X_1, has a density p with respect to the Lebesgue measure on $\mathbb{R}$. The fact that X_1 has a light tail is captured in the hypothesis that X_1 has a moment generating function

$$\Phi(t) := E[\exp tX_1],$$

which is finite in a non void neighborhood $\mathcal{N}$ of 0. This fact is usually refered to as a Cramer type condition.

Defined on $\mathcal{N}$ are the following functions. The functions

$$t \to m(t) := \frac{d}{dt} \log \Phi(t) \tag{V.64}$$

$$t \to s^2(t) := \frac{d}{dt} m(t) \tag{V.65}$$

$$t \to \mu_j(t) := \frac{d^j}{dt^j} \log \Phi(t),\ j \geq 3 \tag{V.66}$$

are the expectation, the variance, and the centered moments of order j of the r.v. $\mathcal{X}_t$ with density

$$\pi_t(x) := \frac{\exp tx}{\Phi(t)} p(x)$$

which is defined on $\mathbb{R}$ and which is the tilted density with parameter t. When Φ is steep, meaning that

$$\lim_{t \to t^+} m(t) = \infty$$

where $t^+ := ess \sup \mathcal{N}$ then m parametrizes the convex hull of the support of P_X. We refer to Barndorff-Nielsen **?** for those properties. As a consequence of this fact, for all a in the support of P_X, it will be convenient to define

$$\pi^a = \pi_t$$

where a is the unique solution of the equation $m(t) = a$.

The Gibbs conditional principle in the standard above setting can be stated as follows.

As n tends to infinity the conditional distribution of X_1 given $(S_1^n/n > a)$ is Π^a, the distribution with density π^a.

Indeed we prefer to state Gibbs principle in a form where the conditioning event is a point condition $(S_1^n/n = a)$. The conditional distribution of X_1 given $(S_1^n/n = a)$ is a well defined distribution and Gibbs conditional principle states that this conditional distribution converges to Π^a as n tends to infinity. In both settings, this convergence holds in total variation norm. We refer to Diaconis and Freedman (1988) for the local form of the conditioning event ; we will mostly be interested in the extension of this form in the present paper.

For all α (depending on n or not) we will denote p_α the density of the random vector X_1^k conditioned upon the local event $(S_1^n = n\alpha)$. The notation $p_\alpha\left(X_1^k = x_1^k\right)$ is sometimes used to denote the value of the density p_α at point x_1^k. The same notation is used when $X_1, \ldots, X_k$ are sampled under some Π^α, namely $\pi^\alpha(X_1^k = x_1^k)$.

This article is organized as follows. Notation and hypotheses are stated in Section 2.2, along with some necessary facts from asymptotic analysis in the context of light tailed densities. Section 2.4 provides a local Gibbs conditional principle under EDP, namely producing the approximation of the conditional density of X_1 conditionally on $(S_1^n/n = a_n)$ for sequences a_n which tend to infinity. We explore two rates of growth for the sequence a_n, which yield two different approximating distributions for the conditional law of X_1. The first one extends the classical approximation by the tilted one, substituting π^a by π^{a_n}. The second case, which corresponds to a faster growth of a_n, produces an approximation of a different kind. It may be possible to explore faster growth conditions than those considered here, leading to a wide class of approximating distributions ; this would require some high order Edgeworth expansions for triangular arrays of variables, extending the corresponding result of order 3 presented in this paper ; we did not move further in this direction, in order to avoid too many technicalities.

For fixed k and fixed $a_n = a > E(X_1)$ it is known that the r.v's $X_1, \ldots, X_k$ are asymptotically independent given $(S_1^n/n = a_n)$; see Diaconis and Freedman (1988). This statement is explored when a_n grows to infinity with n, keeping k fixed. It is shown that the asymtotic independence property holds for sequences a_n with moderate growth, and that independence fails for sequences a_n with fast growth.

The local approximation of the density of X_1 conditionally on $(S_1^n/n = a_n)$ is further extended to typical paths under the conditional sampling scheme, which in turn provides the approximation in variation norm for the conditional distribution ; the method used here follows closely the approach by **?**. The differences between the Gibbs principles in LDP and EDP are discussed. Section 2.5 states similar results in the case when the conditioning event is

$(S_1^n/n > a_n)$.

The main tools to be used come from asymptotic analysis and local limit theorems, developped from Feller (1971) and Bingham et al. (1987); we also have borrowed a number of arguments from **?**. An Edgeworth expansion for some special array of independent r.v's with tilted distribution and argument moving to infinity with the row-size is needed; its proof is differed to the Section 2.6. The basic Abelian type result which is used is stated in **?**.

2.2 Notation and hypotheses

Thereafter we will use indifferently the notation $f(t) \underset{t\to\infty}{\sim} g(t)$ and $f(t) \underset{t\to\infty}{=} g(t)(1+o(1))$ to specify that f and g are asymptotically equivalent functions.

The density p is assumed to be of the form

$$p(x) = \exp(-(g(x) - q(x))), \quad x \in \mathbb{R}_+. \tag{V.67}$$

The function q is assumed to be bounded, so that the asymptotic behaviour of p is captured through the function g. The function g is positive, convex, four times differentiable and satisfies

$$\frac{g(x)}{x} \underset{x\to\infty}{\longrightarrow} \infty. \tag{V.68}$$

Define

$$h(x) := g'(x). \tag{V.69}$$

In the present context, due to (V.68) and the assumed conditions on q to be stated hereunder, $t^+ = +\infty$.

Not all positive convex g's satisfying (V.68) are adapted to our purpose. We follow the line of Juszczak and Nagaev **?** to describe the assumed regularity conditions of h. See also **?** for somehow similar conditions.

We firstly assume that the function h, which is a positive function defined on $\mathbb{R}_+$, is either regularly or rapidly varying in a neighborhood of infinity; the function h is monotone and, by (V.68), $h(x) \to \infty$ when $x \to \infty$.

The following notation is adopted.

$RV(\alpha)$ designates the class of regularly varying functions of index α defined on $\mathbb{R}_+$,

$$\psi(t) := h^{\leftarrow}(t)$$

designates the inverse of h. Hence ψ is monotone for large t and $\psi(t) \to \infty$ when $t \to \infty$, $\sigma^2(x) := 1/h'(x)$, $\hat{x} := \hat{x}(t) = \psi(t)$, $\hat{\sigma} := \sigma(\hat{x}) = \sigma(\psi(t))$.

The two cases considered for h, the regularly varying case and the rapidly varying case, are described below. The first one is adapted to regularly varying functions g, whose smoothness is described through the following condition pertaining to h.

The Regularly varying case. It will be assumed that h belongs to the subclass of $RV(\beta)$, $\beta > 0$, with

$$h(x) = x^\beta l(x),$$

where the Karamata form of the slowly varying function l takes the form

$$l(x) = c \exp \int_1^x \frac{\epsilon(u)}{u} du \tag{V.70}$$

for some positive c. We assume that $x \mapsto \epsilon(x)$ is twice differentiable and satisfies

$$\begin{cases} \epsilon(x) \underset{x\to\infty}{=} o(1), \\ x|\epsilon'(x)| \underset{x\to\infty}{=} O(1), \\ x^2|\epsilon^{(2)}(x)| \underset{x\to\infty}{=} O(1). \end{cases} \tag{V.71}$$

It will also be assumed that

$$|h^{(2)}(x)| \in RV(\theta) \tag{V.72}$$

where θ is a real number such that $\theta \leq \beta - 2$.

Remark 6. *Under (V.70), when $\beta \neq 1$ then, under (V.72), $\theta = \beta - 2$. Whereas, when $\beta = 1$ then $\theta \leq \beta - 2$. A sufficient condition for the last assumption (V.72) is that $\epsilon'(t) \in RV(\gamma)$, for some $\gamma < -1$. Also in this case when $\beta = 1$, then $\theta = \beta + \gamma - 1$.*

Example 12. Weibull density. *Let p be a Weibull density with shape parameter $k > 1$ and scale parameter 1, namely*

$$\begin{aligned} p(x) &= kx^{k-1} \exp(-x^k), \qquad x \geq 0 \\ &= k \exp(-(x^k - (k-1)\log x)). \end{aligned}$$

Take $g(x) = x^k - (k-1)\log x$ and $q(x) = 0$. Then it holds

$$h(x) = kx^{k-1} - \frac{k-1}{x} = x^{k-1}\left(k - \frac{k-1}{x^k}\right).$$

Set $l(x) = k - (k-1)/x^k, x \geq 1$, *which verifies*

$$l'(x) = \frac{k(k-1)}{x^{k+1}} = \frac{l(x)\epsilon(x)}{x}$$

with

$$\epsilon(x) = \frac{k(k-1)}{kx^k - (k-1)}.$$

Since the function $\epsilon(x)$ *satisfies the three conditions in (V.71), then* $h(x) \in RV(k-1)$.

The Rapidly varying case. Here we have $h^{\leftarrow}(t) = \psi(t) \in RV(0)$ and

$$\psi(t) = c \exp \int_1^t \frac{\epsilon(u)}{u} du, \tag{V.73}$$

for some positive c, and $t \mapsto \epsilon(t)$ is twice differentiable with

$$\begin{cases} \epsilon(t) \underset{t\to\infty}{=} o(1), \\ \frac{t\epsilon'(t)}{\epsilon(t)} \underset{t\to\infty}{\longrightarrow} 0, \\ \frac{t^2\epsilon^{(2)}(t)}{\epsilon(t)} \underset{t\to\infty}{\longrightarrow} 0. \end{cases} \tag{V.74}$$

Note that these assumptions imply that $\epsilon(t) \in RV(0)$.

Example 13. A rapidly varying density. *Define* p *through*

$$p(x) = c \exp(-e^{x-1}), x \geq 0.$$

Then $g(x) = h(x) = e^{x-1}$ *and* $q(x) = 0$ *for all non negative* x. *We show that* $h(x)$ *is a rapidly varying function. It holds* $\psi(t) = \log t + 1$. *Since* $\psi'(t) = 1/t$, *let* $\epsilon(t) = 1/(\log t + 1)$ *so that* $\psi'(t) = \psi(t)\epsilon(t)/t$. *Moreover, the three conditions of (V.74) are satisfied. Thus* $\psi(t) \in RV(0)$ *and* $h(x)$ *is a rapidly varying function.*

Denote by $\mathcal{R}$ the class of functions with either regular variation defined as in Case 2.2 or with rapid variation defined as in Case 2.2.

We now state hypotheses pertaining to the bounded function q in (V.67). We assume that

$$|q(x)| \in RV(\eta), \text{ for some } \eta < \theta - \tfrac{3\beta}{2} - \tfrac{3}{2} \text{ if } h \in RV(\beta) \tag{V.75}$$

and

$$|q(\psi(t))| \in RV(\eta), \text{ for some } \eta < -\tfrac{1}{2} \text{ if } h \text{ is rapidly varying.} \tag{V.76}$$

We will make use of the following result (see **?** Thm 3.1).

Theorem 8. *Let $p(x)$ be defined as in (V.67) and $h(x)$ belong to $\mathcal{R}$. Denote by $m(t)$, $s^2(t)$ and $\mu_j(t)$ for $j = 3, 4, ...$ the functions defined in (V.64), (V.65) and (V.66). Then it holds*

$$\begin{aligned}
m(t) &\underset{t\to\infty}{=} \psi(t)(1+o(1)),\\
s^2(t) &\underset{t\to\infty}{=} \psi'(t)(1+o(1)),\\
\mu_3(t) &\underset{t\to\infty}{=} \psi^{(2)}(t)(1+o(1)),\\
\mu_j(t) &\underset{t\to\infty}{=} \begin{cases} M_j s^j(t)(1+o(1)), & \text{for even } j>3\\ \frac{(M_{j+3}-3jM_{j-1})\mu_3(t)s^{j-3}(t)}{6}(1+o(1)), & \text{for odd } j>3 \end{cases},
\end{aligned}$$

where M_i, $i > 0$, denotes the ith order moment of standard normal distribution.

Corollary 4. *Let $p(x)$ be defined as in (V.67) and $h(x) \in \mathfrak{R}$. Then it holds as $t \to \infty$*

$$\frac{\mu_3(t)}{s^3(t)} \longrightarrow 0.$$

Démonstration. In the regularly varying case this follows from Corollaries 1 and 2 in (Biret et al., 2015), and in the rapidly varying case from Corollary 3 and Lemma 3 in (Biret et al., 2015). ■

Our results require an extension of the classical Edgeworth expansions to triangular arrays of row-wise independent and identically distributed random variables, where the expectation of the generic r.v. in the $n-$th row tends to infinity with n. This can be achieved under log-concavity of p, i.e. when the function q is the null function, or when p is nearly log-concave. This is the scope of the next Section.

2.3 Edgeworth expansion under extreme normalizing factors

With π^{a_n} defined through

$$\pi^{a_n}(x) = \frac{e^{tx}p(x)}{\Phi(t)},$$

and t determined by $m(t) = a_n$ together with $s^2 := s^2(t)$ define the normalized density of π^{a_n} by

$$\bar{\pi}^{a_n}(x) = s\pi^{a_n}(sx + a_n),$$

and denote the n-convolution of $\bar{\pi}^{a_n}(x)$ by $\bar{\pi}_n^{a_n}(x)$. Denote by ρ_n its normalized density

$$\rho_n(x) := \sqrt{n}\bar{\pi}_n^{a_n}(\sqrt{n}x).$$

The following result extends the local Edgeworth expansion of the distribution of normalized sums of i.i.d. r.v.'s to the present context, where the summands are generated under the density $\bar{\pi}^{a_n}$. Therefore the setting is that of a triangular array of rowwise independent summands ; the fact that $a_n \to \infty$ makes the situation unusual. We mainly adapt Feller's proof (Chapiter 16, Theorem 2 (Feller, 1971)). However this variation on the classical Edgeworth expansion result requires some additional regularity assumption, which meet the requirements of Theorem 8, which are fulfilled in most models dealing with extremes and convolutions. Those are captured in cases when the density p is log-concave, or nearly log concave in the upper tail. Similar conditions are considered in Broniatowski and Cao (2012).

Theorem 9. *With the above notation, uniformly upon x it holds*

$$\rho_n(x) = \phi(x)\Big(1 + \frac{\mu_3}{6\sqrt{n}s^3}\big(x^3 - 3x\big)\Big) + o\Big(\frac{1}{\sqrt{n}}\Big).$$

where $\phi(x)$ is standard normal density.

The proof of this result is postponed to the Section 2.6.

2.4 Gibbs' conditional principles under extreme events

We now explore Gibbs conditional principles under extreme events. The first result is a pointwise approximation of the conditional density $p_{a_n}(y_1)$ on $\mathbb{R}$. Two cases will be considered according to the rate of growth of the sequence a_n to infinity. For "moderate" growth our result extends the classical one pertaining to constant a_n larger than $E(X_1)$, since the approximating density of p_{a_n} is the tilted distribution with parameter a_n. For sequences a_n with fast growth, the approximating density includes a second order term which contributes to the approximation in a similar role as the tilted term ; this term also appears in the first case, but is negligible with respect to the tilted density.

However this local approximation can be greatly improved when comparing p_{a_n} to its approximation. We will first prove that the approximation holds when the fixed arbitrary y_1 is substituted by a r.v. Y_1 with distribution P_{a_n}, henceforth on a typical realization under the distribution to be approximated. The approximation therefore holds in probability under this

sampling scheme; a simple Lemma then proves that such a statement implies that the total variation distance between P_{a_n} and its approximation tends to 0 as n tends to infinity.

As a by-product we also address similar approximations for the case when the conditioning event writes $(S_1^n/n > a_n)$. The case when a_n grows to infinitly fast enough overlaps with that for which (V.63) holds.

Extension to the approximation of the distribution of X_1 given $(T_n = a_n)$ or $(T_n > a_n)$ where

$$T_n := \frac{1}{n}\sum_{i=1}^{n} f(X_i)$$

for functions f satisfying appropriate conditions are considered.

For sake of completeness we also provide some information when the density p_{a_n} is that of the vector $(X_1, \ldots, X_k)$ for fixed k. We prove that for moderate growth of a_n the approximating density is the product of corresponding marginal approximations, generalizing the well known result by Csiszar (Csiszár, 1984) which, in the present context, assesses the limit conditional independence of the coordinates of the vector $(X_1, \ldots, X_k)$ given $(S_n > na_n)$ for fixed $a_n > E(X_1)$ and fixed k. At the contrary this property is lost when a_n grows quickly to infinity.

Because of the property (V.63) it would be of interest to consider the joint distribution of the vector $(X_1, \ldots, X_{k_n})$ given $(S_n > na_n)$ for sequences k_n close to n, as done in (Broniatowski et al., 2014) for sequences a_n ranging from CLT to LDP. The extreme deviation case adds noticeable analytical difficulties.

2.4.1 A local result

Fix $y_1^k := (y_1, \ldots, y_k)$ in $\mathbb{R}^k$ and define $s_i^j := y_i + \ldots + y_j$ for $1 \le i < j \le k$. Define t through

$$m := m(t) := a_n \tag{V.77}$$

and set

$$s := s(t)$$

for brevity.

We consider two conditions pertaining to the growth of the sequence a_n to infinity. In the first case we assume that

$$\lim_{n\to\infty} \frac{a_n}{s\sqrt{n}} = 0, \tag{V.78}$$

and in the second case we consider sequences a_n which may grow faster to infinity, obbeying

$$0 < \lim\inf_{n\to\infty} \frac{a_n}{s\sqrt{n}} \leq \lim\sup_{n\to\infty} \frac{a_n}{s\sqrt{n}} < \infty. \tag{V.79}$$

Remark 7. *Both conditions (V.78) and (V.79) can be expressed in terms of a_n when the variance function $V(x)$ of the distribution of X_1 is known either in closed form, or is asymptotically equivalent to some known function. Recall that the variance function is defined on $Im(X_1)$ through*

$$x \to V(x) = s^2 om^{-1}(x).$$

See e.g. (Bar-Lev et al., 1992) for a description of distribution functions with polynomial variance function and (Jørgensen et al., 1997) for tail equivalence for the variance function in infinitely divisible distributions. In the Regularly varying case, i.e. when h belongs to the subclass of $RV(\beta)$, $\beta > 0$, then standard operations on smooth regularly varying functions yield $V(x) = x^{1-\beta}\ l(x)$ for some slowly varying function l ; see (Bingham et al., 1987) ; hence $s(t) = a_n^{(1-\beta)/2} l(a_n)$. Assuming that $V(x) \sim x^{2\rho}$ as $x \to \infty$, it follows that (V.78) writes $a_n = o\left(n^{1/(1+\rho)}\right)$ whereas (V.79) amounts to assume that a_n is of order $n^{1/(1+\rho)}$.

Denote

$$z := \frac{m - y_1}{s\sqrt{n-1}}.$$

Theorem 10. *When (V.78) holds then it holds*

$$p_{a_n}(y_1) = p(X_1 = y_1 | S_1^n = na_n) = \pi^{a_n}(y_1)\Big(1 + o\Big(\frac{1}{\sqrt{n}}\Big)\Big),$$

The proof of this result is postponed to the Section 2.6.

Remark 8. *The above condition (V.78) is not sufficient to entail (V.63) to hold. This yields to the study of a similar limit conditional result under the corresponding condition (V.79).*

Theorem 11. *Assume that the sequence a_n satisfies (V.79). Denote*

$$\alpha := t + \frac{\mu_3}{2(n-1)s^2}$$

and

$$\beta := (n-1)s^2.$$

Then

$$p(X_1 = y_1 | S_1^n = na_n) = g_{a_n}(y_1) := Cp(y_1)\mathfrak{n}\left(\alpha\beta + a_n, \beta, y_1\right)(1 + o(1)) \tag{V.80}$$

where $\mathfrak{n}(\mu, \sigma^2, x)$ *denotes the normal density with expectation* μ *and variance* σ^2 *evaluated at point* x*, and* C *is a normalizing constant.*

When (V.78) holds instead of (V.79) then

$$p(X_1 = y_1 | S_1^n = na_n) = \pi^{a_n}(y_1)(1 + o(1))$$

for all y_1 *as* n *tends to infinity.*

Démonstration. In contrast with the above case, the second summand in (V.101) does not tend to 0 any longer and contributes to the approximating density. Standard development then yields the result. When (V.78) holds instead of (V.79) then standard expansions in (V.80) provide $g_{a_n}(y_1) \sim \pi^{a_n}(y_1)$ for all y_1 as n tends to infinity. ■

2.4.2 On conditional independence under extreme events

We now turn to the case when we approximate the joint conditional density $p_{a_n}(y_1^k) := p_{a_n}(y_1, \ldots, y_k) = p_{a_n}(X_1^k = y_1^k \big| S_1^n = na_n)$. Denote $s_i^j := y_i + \ldots + y_j$ for $i \leq j$ and $s_1^0 := 0$.

We first consider the case when (V.78) holds. We then have

Proposition 1. *When (V.78) holds then for any fixed* k

$$p_{a_n}(y_1^k) = \prod_{i=1}^{k} \pi^{m_i}(y_i)\Big(1 + o(1/\sqrt{n})\Big)$$

where

$$m_i := m(t_i) := \frac{na_n - s_1^i}{n - i}. \tag{V.81}$$

The proof of this result is postponed to the Section 2.6.

We now explore the limit conditional independence of blocks of fixed length under extreme condition. As a consequence of the above Proposition 1 it holds

Theorem 12. *Under (V.78) it holds*

$$p_{a_n}(y_1^k) = p(X_1^k = y_1^k | S_1^n = na_n) = \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right) \prod_{i=1}^{k} \pi^{a_n}(X_i = y_i),$$

The technical proof is differed to the Section 2.6.

Remark 9. *The above result shows that asymptotically the point condition* $(S_1^n = na_n)$ *leaves blocks of* k *of the* X_i'*s independent. Obviously this property does not hold for large values of*

k, close to n. A similar statement holds in the LDP range, conditioning either on $(S_1^n = na)$ *(see Diaconis and Friedman (Diaconis and Freedman, 1988))), or on* $(S_1^n \geq na)$ *(see Csiszar (Csiszár, 1984) for a general statement on asymptotic conditional independence given events with positive probability).*

We now turn to the case when a_n moves more quickly to infinity. Denote

$$m_i := m(t_i) := \frac{na_n - s_1^{i-1}}{n - i + 1}$$

together with

$$s_i^2 := s^2(t_i).$$

Theorem 13. *Assume that (V.79) holds. Then for all fixed k it holds*

$$p(X_1^k = y_1^k | S_1^n = na_n) = \prod_{i=1}^{k} g_i(y_i) \left(1 + o\left(1\right)\right)$$

where

$$g_i(y_i) := C_i p(y_i) \mathfrak{n}\left(\alpha_i \beta_i + a_n, \beta_i, y_i\right)$$

and

$$\alpha_i := t_i + \frac{\mu_3}{2(n - i + 1)s_i^2} \tag{V.82}$$

$$\beta_i := (n - i + 1)s_i^2. \tag{V.83}$$

Remark 10. *When (V.78) holds, the above result is a refinement of the result in Proposition 1. Under (V.79) and when (V.78) does not hold, the approximations obtained in Lemma 15 do not hold, and the approximating density cannot be stated as a product of densities under which independence holds. In that case it follows that the conditional independence property under extreme events does not hold any longer.*

2.4.3 Strenghtening the local Gibbs conditional principle

We now turn to a stronger approximation of p_{a_n}. Consider Y_1 a r.v. with density p_{a_n}, and the random variable $p_{a_n}(Y_1) := p(X_1 = Y_1 | S_1^n = na_n)$. Denote

$$g_{a_n}(x) := Cp(x) \mathfrak{n}\left(\alpha\beta + a_n, \beta, x\right)$$

where $\alpha := \alpha_n$ and $\beta := \beta_n$ are defined in (V.82) and (V.83), and C is a normalizing constant, it holds

Theorem 14. *(i) When (V.78) holds then*

$$p_{a_n}(Y_1) = \pi^{a_n}(Y_1)(1 + R_n)$$

where the tilted density at point a_n, and where R_n is a function of Y_1 such that $P_{a_n}(|R_n| > \delta\sqrt{n}) \to 0$ as $n \to \infty$ for any positive δ. When (V.79) holds then, with t_n such that $m(t_n) = a_n$, $\alpha := \alpha_n$ and $\beta := \beta_n$

$$p_{a_n}(Y_1) = g_{a_n}(Y_1)(1 + R'_n)$$

where $P_{a_n}(|R'_n| > \delta) \to 0$ as $n \to \infty$ for any positive δ.

Remark 11. *This result is of much greater relevance than the previous ones. Indeed under P_{a_n} the r.v. Y_1 may take large values. On the contrary simple approximation of p_{a_n} by π^{a_n} or g_{a_n} on $\mathbb{R}_+$ only provides some knowledge on p_{a_n} on sets with smaller and smaller probability under p_{a_n}. Also it will be proved that as a consequence of the above result, the L^1 norm between p_{a_n} and its approximation goes to 0 as $n \to \infty$, a result out of reach through the aforementioned results.*

In order to adapt the proof of Theorem 12 to the present setting it is necessary to get some insight on the plausible values of Y_1 under P_{a_n}. It holds

Lemma 10. *It holds*

$$Y_1 = O_{P_{a_n}}(a_n).$$

Démonstration. This is a consequence of Markov Inequality :

$$P(Y_1 > u | S_1^n = na_n) \leq \frac{E(Y_1 | S_1^n = na_n)}{u} = \frac{a_n}{u}$$

which goes to 0 for all $u = u_n$ such that $\lim_{n\to\infty} u_n/a_n = \infty$.

Now making use of Lemma 10 in the proof of Theorem 10 and Theorem 11, substituting y_1 with Y_1, completes the proof. ■

Denote the probability measures P_{a_n} , Π^{a_n} and G_{a_n} with respective densities p_{a_n}, π^{a_n} and g_{a_n} .

2.4.4 Gibbs principle in variation norm

We now consider the approximation of P_{a_n} by G_{a_n} in variation norm.

The main ingredient is the fact that in the present setting approximation of p_{a_n} by g_{a_n} in probability plus some rate implies approximation of the corresponding measures in variation norm. This approach has been developped in (Broniatowski et al., 2014) ; we state a first lemma which states that wether two densities are equivalent in probability with small relative error when measured according to the first one, then the same holds under the sampling of the second.

Let $\mathfrak{R}_n$ and $\mathfrak{S}_n$ denote two p.m's on $\mathbb{R}^n$ with respective densities $\mathfrak{r}_n$ and $\mathfrak{s}_n$.

Lemma 11. *Suppose that for some sequence ϖ_n which tends to 0 as n tends to infinity*

$$\mathfrak{r}_n\left(Y_1^n\right) = \mathfrak{s}_n\left(Y_1^n\right)\left(1 + o_{\mathfrak{R}_n}(1)\right) \tag{V.84}$$

as n tends to ∞. Then

$$\mathfrak{s}_n\left(Y_1^n\right) = \mathfrak{r}_n\left(Y_1^n\right)\left(1 + o_{\mathfrak{S}_n}(1)\right). \tag{V.85}$$

The proof of this result is available in (Broniatowski et al., 2014). Applying this Lemma to the present setting yields

$$g_{a_n}\left(Y_1\right) = p_{a_n}\left(Y_1\right)\left(1 + o_{G_{a_n}}\left(1/\sqrt{n}\right)\right)$$

as $n \to \infty$, which together with Theorem 11 or Theorem 10 implies

$$p_{a_n}\left(Y_1\right) = g_{a_n}\left(Y_1\right)\left(1 + o_{P_{a_n}}\left(1/\sqrt{n}\right)\right)$$

or

$$p_{a_n}\left(Y_1\right) = \pi^{a_n}\left(Y_1\right)\left(1 + o_{P_{a_n}}\left(1/\sqrt{n}\right)\right)$$

This fact entails

Theorem 15. *Under (V.79) the total variation norm between P_{a_n} and G_{a_n} goes to 0 as $n \to \infty$. When (V.78) holds then the total variation norm between P_{a_n} and Π^{a_n} goes to 0 as $n \to \infty$.*

The proof of this theorem is also provided in (Broniatowski et al., 2014).

Remark 12. *This result is to be paralleled with Theorem 1.6 in Diaconis and Freedman (Diaconis and Freedman, 1988) and Theorem 2.15 in Dembo and Zeitouni (Dembo and Zeitouni, 1996) which provide a rate for this convergence in the LDP range.*

2.4.5 The asymptotic location of X under the conditioned distribution

This paragraph intends to provide some insight on the behaviour of X_1 under the condition $(S_1^n = na_n)$; this will be extended further on to the case when $(S_1^n \geq na_n)$ and to be considered in parallel with similar facts developped in (Broniatowski et al., 2014) for larger values of a_n.

Let $\mathcal{X}_t$ be a r.v. with density π^{a_n} where $m(t) = a_n$ and a_n satisfies (V.78) or (V.79). Recall that $E\mathcal{X}_t = a_n$ and $Var\mathcal{X}_t = s^2$. We evaluate the moment generating function of the normalized variable $(\mathcal{X}_t - a_n)/s$. It holds

$$\log E[\exp(\lambda(\mathcal{X}_t - a_n)/s)] = -\lambda a_n/s + \log\Phi\left(t + \frac{\lambda}{s}\right) - \log\Phi(t).$$

A second order Taylor expansion in the above display yields

$$\log E[\exp(\lambda(\mathcal{X}_t - a_n)/s)] = \frac{\lambda^2}{2}\frac{s^2\left(t + \frac{\theta\lambda}{s}\right)}{s^2}$$

where $\theta = \theta(t,\lambda) \in (0,1)$. The proof of the following Lemma is deferred to the Section 2.6. It holds

Lemma 12. *Under the above hypotheses and notation, for any compact set K,*

$$\lim_{n\to\infty}\sup_{u\in K}\frac{s^2\left(t + \frac{u}{s}\right)}{s^2} = 1.$$

Applying the above Lemma it follows that the normalized r.v's $(\mathcal{X}_t - a_n)/s$ converge to a standard normal variable $N(0,1)$ in distribution, as $n \to \infty$. This amounts to say that

$$\mathcal{X}_t = a_n + sN(0,1) + o_{\Pi^{a_n}}(1).$$

which implies that $\mathcal{X}_t$ concentrates around a_n with rate s. Due to Theorem 15 the same holds for X_1 under $(S_1^n = na_n)$.

2.4.6 Conditional limit behaviour under other mean effect events

Let $X_1, \ldots, X_n$ denote n i.i.d. real valued r.v's with distribution P and density p and let $f : \mathbb{R} \to \mathbb{R}$ be a measurable function such that $\Phi_f(\lambda) := E[\exp(\lambda f(X_1))]$ is finite for λ in a non void neighborhood of 0 (the so-called Cramer condition). Denote $m_f(\lambda)$ and $s_f^2(\lambda)$ the first and second derivatives of $\log\Phi_f(\lambda)$. Assume that the r.v. $f(X_1)$ has density p_f on $\mathbb{R}$, and denote $p_f(f(X_1) = u)$ its value at point u.

Denote

$$\pi_f^a(y) = \frac{\exp \lambda y}{\Phi_f(\lambda)} p_f(y)$$

with λ the unique solution of the equation $m_f(\lambda) = a$ for all a in $Im(f(X_1))$ assuming that $\lambda \to \Phi_f(\lambda)$ is steep on its domain. Denote

$$F_i^j := f(X_i) + \ldots + f(X_j)$$

for $1 \leq i \leq j \leq n$. We make use of the following equality

$$\begin{aligned} p(X_1 = x | F_1^n = na_n) \\ &= \frac{p(X_1 = x)}{p_f(f(X_1) = f(x))} \times \\ &\left(p_f(f(X_1) = f(x)) \frac{p_f(f(X_1) = f(x) | F_1^n = na_n)}{p_f(f(X_1) = f(x) | F_2^n = na_n - f(x))} \right). \end{aligned}$$

Note that for all α in $Im(f(X_1))$, denoting λ the solution of $m_f(\lambda) = \alpha$ and defining

$${}_f\pi^\alpha(x) := \frac{e^{\lambda f(x)} p(X = x)}{\int e^{\lambda f(x)} p(X = x) dx}$$

it is readily checked that

$${}_f\pi^\alpha(x) = \frac{p(X_1 = x)}{p_f(f(X_1) = f(x))} \pi_f^{a_n}(f(x)).$$

Denoting $P_{a_n,f}$ the distribution of X_1 given $(F_1^n = na_n)$ it results, using Theorem 14 that the following Theorem holds.

Theorem 16. *Assume that, with s substituted by s_f, condition (V.78) holds. Then*

$$p(X_1 = Z | F_1^n = na_n) =_f \pi^\alpha(Z) \left(1 + o_{P_{a_n,f}}\left(1/\sqrt{n}\right)\right)$$

and under (V.79)

$$\begin{aligned} p(X_1 = Z | F_1^n &= na_n) \\ &= C \frac{p(X_1 = Z)}{p_f(f(X_1) = f(Z))} \mathfrak{n}(\alpha\beta + a_n, \beta, f(Z)) \left(1 + o_{P_{a_n,f}}\left(1/\sqrt{n}\right)\right) \end{aligned}$$

where $\alpha := \alpha_n$ and $\beta := \beta_n$ are defined in (V.82) and (V.83) with m and s substituted by m_f

and s_f.

Remark 13. *The first part of the above Theorem extends the classical Gibbs Principle under condition (V.78), which, for fixed* $a = a_n$ *writes*

$$p(X_1 = x | F_1^n = na) =_f \pi^\alpha(x) \left(1 + o\left(1/\sqrt{n}\right)\right)$$

for any fixed x. *See (Diaconis and Freedman, 1988). This statement does not hold any longer under condition (V.79).*

Remark 14. *Making use of the same arguments as in Subsection 2.4.4 it follows that Theorem 16 yields that the variation distance between the conditional distribution and its approximation tends to* 0 *as* n *tends to infinity.*

Example 14. *Consider for example the application of the above result to r.v's* $Y_1, \ldots, Y_n$ *with* $Y_i := (X_i)^2$ *where the* $X_i's$ *are i.i.d. and are such that the density of the i.i.d. r.v's* $Y_i's$ *satisfy (V.67), where* $h \in R_\beta \cup R_\infty$ *with* $\beta > 1$. *By the Gibbs conditional principle, for fixed* a, *conditionally on* $(\sum_{i=1}^n Y_i = na)$ *the generic r.v.* Y_1 *has a non degenerate limit distribution*

$$p_Y^*(y) := \frac{\exp ty}{E \exp tY_1} p_Y(y)$$

and the limit density of X_1 *under* $(\sum_{i=1}^n X_i^2 = na)$ *is*

$$p_X^*(y) := \frac{\exp tx^2}{E \exp tX_1^2} p_X(y)$$

whereas, when $a_n \to \infty$, Y_1*'s the limit conditional distribution is degenerate and concentrates around* a_n. *As a consequence the distribution of* X_1 *under the condition* $(\sum_{i=1}^n X_i^2 = na_n)$ *concentrates sharply at* $-\sqrt{a_n}$ *and* $+\sqrt{a_n}$.

2.5 EDP under exceedance

The following proposition states the marginally conditional density under condition $A_n = \{S_1^n \geq na_n\}$. We denote this density by p_{A_n} to differentiate it from p_{a_n} which is under condition $\{S_1^n = na_n\}$. For the purpose of the proof, we need the following Lemma, based on Theorem 6.2.1 of Jensen (Jensen, 1995) in order to provide the asymptotic estimation of the tail probability $P(S_1^n \geq na_n)$ and of the n-convolution density $p(S_1^n/n = u)$ for $u > a_n$.

Define

$$I(x) := xm^{-1}(x) - \log \Phi\left(m^{-1}(x)\right).$$

We make use of the following result (see Section 2.6 for the proof).

Lemma 13. *Set $m(t) = a_n$. Suppose that $a_n \to \infty$ as $n \to \infty$. Then it holds*

$$P(S_1^n \geq na_n) = \frac{\exp(-nI(a_n))}{\sqrt{2\pi}\sqrt{n}ts(t)} \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right). \tag{V.86}$$

Let further t_τ be defined by $m(t_\tau) = \tau$ with $\tau \geq a_n$, it then holds, uniformly upon τ

$$p(S_1^n = n\tau) = \frac{\sqrt{n}\exp(-nI(\tau))}{\sqrt{2\pi}s(t_\tau)} \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right). \tag{V.87}$$

The proof of this lemma is postponed to the Section 2.6.

Theorem 17. *Let $X_1, \ldots, X_n$ be i.i.d. random variables with density $p(x)$ defined in (V.67) and $h(x) \in \mathcal{R}$. Set $m(t) = a_n$ let η_n be a positive sequence satisfying*

$$\eta_n \longrightarrow 0 \qquad \text{and} \qquad nm^{-1}(a_n)\eta_n \longrightarrow \infty.$$

(i) When (V.78) holds

$$p_{A_n}(y_1) = p(X_1 = y_1 | S_1^n \geq na_n) = \pi_{A_n}(y_1)\left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right),$$

with

$$\pi_{A_n}(y_1) = ts(t)e^{nI(a_n)} \int_{a_n}^{a_n+\eta_n} \pi_\tau(y_1) \exp\left(-nI(\tau) - \log s(t_\tau)\right) d\tau$$

with t_τ defined by $m(t_\tau) = \tau$.

(ii) When (V.79) holds

$$p_{A_n}(y_1) = p(X_1 = y_1 | S_1^n \geq na_n) = g_{A_n}(y_1)\Big(1 + o\Big(\frac{1}{\sqrt{n}}\Big)\Big),$$

with

$$g_{A_n}(y_1) = ts(t)e^{nI(a_n)} \int_{a_n}^{a_n+\eta_n} g_\tau(y_1) \exp\left(-nI(\tau) - \log s(t_\tau)\right) d\tau,$$

where $g_\tau = \pi^\tau$ with t_τ defined by $m(t_\tau) = \tau$.

The proof is postponed to the Section 2.6.

Remark 15. *Conditions (V.79) and (V.78) have to be compared with the growth condition pertaining to the sequence a_n for which (V.63) holds. Consider the regularly varying case, namely*

assume that $h(x) = x^{\beta} l(x)$ *for some* $\beta > 0$. *Then making use of Theorem 8 it is readily checked that (V.79) amounts to*

$$\lim \inf_{n\to\infty} \frac{a_n}{n^{1/(1+\beta)}} > 0. \tag{V.88}$$

Now (V.63) holds whether for some $\delta > 1/(\beta+1)$

$$\lim \inf_{n\to\infty} \frac{a_n}{n^{\delta}} > 0. \tag{V.89}$$

Assume that (V.89) holds; then (V.88) holds for all distributions p *with* $h(x) = x^{\beta} l(x)$ *and* $\beta > (1-\delta)/\delta$. *This can be stated as follows : Assume that for some* $0 < \eta < 1$ *it holds*

$$\lim \inf_{n\to\infty} \frac{a_n}{n^{\eta}} > 0$$

then whenever $\beta > (1-\eta)/\eta$ *(V.89) and (V.88) simultaneously hold.*

2.6 Appendix

2.6.1 Proof of Theorem 9

We state a preliminary Lemma, whose role is to provide some information on the characteristic function of the normalised random variable $(\mathcal{X}_t - m(t))/s(t)$ with density $\widetilde{\pi}_t$ defined by

$$\widetilde{\pi}_t(x) := \frac{s(t) \exp t\,(s(t)x + m(t))\, p(s(t)x + m(t))}{\phi(t)} \tag{V.90}$$

as $t \to \infty$. The density p satisfies the hypotheses in Section 2.2. Denote $\varphi^{a_n}(u) := \int e^{iux} \widetilde{\pi}_t(x) dx$ the characteristic function of $(\mathcal{X}_t - m(t))/s(t)$. It holds

Lemma 14. *Assume that there exists* c_1, c_2 *both positive such that for all* t

$$\widetilde{\pi}_t(x) > c_1 \text{ for } |x| < c_2 \tag{V.91}$$

then under the hypotheses stated in Section 2.2, for any $c > 0$ *there exists* $\rho < 1$ *such that*

$$|\varphi^{a_n}(u)| \leq \rho \tag{V.92}$$

for $|u| > c$ *and all* a_n.

Démonstration. The proof of this Lemma is in (Jensen, 1995), p150; we state it for complete-

ness. Assume (V.91) holds with $\tilde{\pi}_t(x) > c_1$ for $|x| > c_2$ and setting $\epsilon := c_2/2$

$$\begin{aligned} |\varphi^{a_n}(u)| &\leq \left| \int e^{izu} 1\left(|z| < \epsilon\right) c_1 dz \right| + \int \left\{ \tilde{\pi}_t(z) - 1\left(|z| < \epsilon\right) c_1 \right\} dz \\ &\leq c_1 \left(2\epsilon\right) \left| \frac{e^{iu\epsilon} - e^{-iu\epsilon}}{2iu\epsilon} \right| + \left\{ 1 - 2\epsilon c_1 \right\} \end{aligned}$$

and the last expression is independent on a_n and is such that for any $c > 0$ there exists $\rho < 1$ such that the expression is less than ρ for $|u| > c$.

For the density function $p(x)$ Theorem 5.4 of Nagaev (Juszczak and Nagaev, 2004) states that the normalized tilted density of $p(x)$, namely, $\tilde{\pi}_t(x)$ has the property

$$\lim_{a_n \to \infty} \sup_{x \in \mathbb{R}} |\tilde{\pi}_t(x) - \varphi(x)| = 0 \tag{V.93}$$

which proves (V.91). ■

We now turn to the Proof of Theorem 9.

Since the proof is based on characteristic function (c.f.) arguments, we will use the following notation, in accordance with the common use in this area, therefore turning from laplace transform notation to characteristic function ones. Recall that we denote $\tilde{\pi}_t$ the normalized conjugate density of $p(x)$. Also ρ_n is the normalized $n-$fold convolution of $\tilde{\pi}_t$. Hence we consider the triangular array whose $n-$th row consists in n i.i.d. copies of a r.v. with standardized density $\tilde{\pi}_t$ and the sum of the row, divided by $\sqrt{n}$, has density ρ_n. The standard Gaussian density is denoted ϕ. The c.f. of $\tilde{\pi}_t$ is denoted φ^{a_n} so that the c.f. of ρ_n is $\left(\varphi^{a_n}(.)\right)^n$, and $m(t) = a_n$.

Step 1 : In this step, we will express the following formula $G(x)$ by its Fourier transform. Let

$$G(x) := \rho_n(x) - \phi(x) - \frac{\mu_3}{6\sqrt{n}s_n^3} \left(x^3 - 3x\right) \phi(x).$$

From

$$\phi(x) := \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-i\tau x} e^{-\frac{1}{2}\tau^2} d\tau,$$

it follows that

$$\phi'''(x) = -\frac{1}{2\pi} \int_{-\infty}^{\infty} (i\tau)^3 e^{-i\tau x} e^{-\frac{1}{2}\tau^2} d\tau.$$

On the other hand

$$\phi'''(x) = -(x^3 - 3x)\phi(x),$$

which gives

$$(x^3 - 3x)\phi(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (i\tau)^3 e^{-i\tau x} e^{-\frac{1}{2}\tau^2} d\tau. \tag{V.94}$$

By Fourier inversion

$$\rho_n(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\tau x}\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n d\tau. \tag{V.95}$$

Using (V.94) and (V.95), we have

$$G(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\tau x}\left(\varphi^{a_n}(\tau/\sqrt{n})^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right) d\tau.$$

Hence it holds

$$\begin{aligned}&\left|\rho_n(x) - \phi(x) - \frac{\mu_3}{6\sqrt{n}s^3}\left(x^3-3x\right)\phi(x)\right| \\ &\leq \frac{1}{2\pi}\int_{-\infty}^{\infty}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau.\end{aligned}$$

Step 2 : In this step, we show that for large n, the characteristic function φ^{a_n} satisfies

$$\int |\varphi^{a_n}(\tau)|^2 d\tau < \infty$$

By Parseval identity

$$\int |\varphi^{a_n}(\tau)|^2 d\tau = 2\pi\int(\widetilde{\pi}_t(x))^2 dx \leq 2\pi \sup_{x\in\mathbb{R}}\widetilde{\pi}_t(x) < \infty.$$

Use (V.93) to conclude the proof.

Step 3 : In this step, we complete the proof by showing that when $n \to \infty$

$$\int_{-\infty}^{\infty}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau = o\left(\frac{1}{\sqrt{n}}\right). \tag{V.96}$$

The LHS in (V.96) is splitted on $|\tau| > \omega\sqrt{n}$ and on $|\tau| \leq \omega\sqrt{n}$. It holds

$$\begin{aligned}&\sqrt{n}\int_{|\tau|>\omega\sqrt{n}}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau \\ &\leq \sqrt{n}\int_{|\tau|>\omega\sqrt{n}}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)\right|^n d\tau + \sqrt{n}\int_{|\tau|>\omega\sqrt{n}}\left|e^{-\frac{1}{2}\tau^2} + \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau \\ &\leq \sqrt{n}\rho^{n-2}\int_{|\tau|>\omega\sqrt{n}}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)\right|^2 d\tau + \sqrt{n}\int_{|\tau|>\omega\sqrt{n}} e^{-\frac{1}{2}\tau^2}\left(1 + \left|\frac{\mu_3\tau^3}{6\sqrt{n}s^3}\right|\right) d\tau. \end{aligned}\tag{V.97}$$

where we used Lemma 14 from the second line to the third one. The first term of the last line

tends to 0 when $n \to \infty$, since

$$\begin{aligned}
&\sqrt{n}\rho^{n-2} \int_{|\tau|>\omega\sqrt{n}} \left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)\right|^2 d\tau \\
&= \exp\left(\frac{1}{2}\log n + (n-2)\log\rho + \log \int_{|\tau|>\omega\sqrt{n}} \left(\varphi^{a_n}(\tau/\sqrt{n})\right)^2 d\tau\right) \longrightarrow 0.
\end{aligned}$$

By Corollary 4 when $n \to \infty$

$$\begin{aligned}
&\sqrt{n} \int_{|\tau|>\omega\sqrt{n}} e^{-\frac{1}{2}\tau^2}\left(1 + \left|\frac{\mu_3\tau^3}{6\sqrt{n}s^3}\right|\right)d\tau \\
&\leq \sqrt{n} \int_{|\tau|>\omega\sqrt{n}} e^{-\frac{1}{2}\tau^2}|\tau|^3 d\tau = \sqrt{n} \int_{|\tau|>\omega\sqrt{n}} \exp\Big\{-\frac{1}{2}\tau^2 + 3\log|\tau|\Big\}d\tau \\
&= 2\sqrt{n}\exp\big(-\omega^2 n/2 + o(\omega^2 n/2)\big) \longrightarrow 0,
\end{aligned}$$

where the second equality holds from, for example, Chapiter 4 of (Bingham et al., 1987). Summing up, when $n \to \infty$

$$\int_{|\tau|>\omega\sqrt{n}} \left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau = o\Big(\frac{1}{\sqrt{n}}\Big).$$

If $|\tau| \leq \omega\sqrt{n}$, it holds

$$\begin{aligned}
&\int_{|\tau|\leq\omega\sqrt{n}} \left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3 e^{-\frac{1}{2}\tau^2}\right| d\tau \\
&= \int_{|\tau|\leq\omega\sqrt{n}} e^{-\frac{1}{2}\tau^2}\left|\left(\varphi^{a_n}(\tau/\sqrt{n})\right)^n e^{\frac{1}{2}\tau^2} - 1 - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\right| d\tau \\
&= \int_{|\tau|\leq\omega\sqrt{n}} e^{-\frac{1}{2}\tau^2}\left|\exp\Big\{n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2\Big\} - 1 - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\right| d\tau. \qquad \text{(V.98)}
\end{aligned}$$

The integrand in the last display is bounded through

$$|e^\alpha - 1 - \beta| = |(e^\alpha - e^\beta) + (e^\beta - 1 - \beta)| \leq (|\alpha - \beta| + \frac{1}{2}\beta^2)e^\gamma,$$

where $\gamma \geq \max(|\alpha|, |\beta|)$; this inequality follows replacing e^α, e^β by their power series, for real or complex α, β. Denote by

$$\gamma(\tau) = \log\varphi^{a_n}(\tau) + \frac{1}{2}\tau^2.$$

Since $\gamma'(0) = \gamma''(0) = 0$, the third order Taylor expansion of $\gamma(\tau)$ at $\tau = 0$ yields

$$\gamma(\tau) = \gamma(0) + \gamma'(0)\tau + \frac{1}{2}\gamma''(0)\tau^2 + \frac{1}{6}\gamma'''(\xi)\tau^3 = \frac{1}{6}\gamma'''(\xi)\tau^3,$$

where $0 < \xi < \tau$. Hence it holds

$$\left|\gamma(\tau) - \frac{\mu_3}{6s^3}(i\tau)^3\right| = \left|\gamma'''(\xi) - \frac{\mu_3}{s_n^3}i^3\right|\frac{\tau^3}{6}.$$

Here γ''' is continuous; thus we can choose ω small enough such that $|\gamma'''(\xi)| < \rho$ for $|\tau| < \omega$. Meanwhile, for n large enough, according to Corollary 4, we have $\mu_3/s^3 \to 0$. Hence it holds for n large enough

$$\left|\gamma(\tau) - \frac{\mu_3}{6s^3}(i\tau)^3\right| \le \Big(|\gamma'''(\xi)| + \rho\Big)\frac{|\tau|^3}{6} < \rho\tau^3. \tag{V.99}$$

Choose ω small enough, such that for n large enough it holds for $|\tau| < \omega$

$$\left|\frac{\mu_3}{6s^3}(i\tau)^3\right| \le \frac{1}{4}\tau^2, \text{ and } \gamma(\tau)| \le \frac{1}{4}\tau^2.$$

For this choice of ω, when $|\tau| < \omega$ we have

$$\max\left(\left|\frac{\mu_3}{6s^3}(i\tau)^3\right|, |\gamma(\tau)|\right) \le \frac{1}{4}\tau^2.$$

Replacing τ by $\tau/\sqrt{n}$, it holds for $|\tau| < \omega\sqrt{n}$, and using (V.99)

$$\begin{aligned}
&\left|n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2 - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\right| \\
&= n\left|\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\Big(\frac{\tau}{\sqrt{n}}\Big)^2 - \frac{\mu_3}{6s^3}\Big(\frac{i\tau}{\sqrt{n}}\Big)^3\right| \\
&= n\left|\gamma\Big(\frac{\tau}{\sqrt{n}}\Big) - \frac{\mu_3}{6s^3}\Big(\frac{i\tau}{\sqrt{n}}\Big)^3\right| < \frac{\rho|\tau|^3}{\sqrt{n}}.
\end{aligned}$$

In a similar way, it also holds for $|\tau| < \omega\sqrt{n}$

$$\begin{aligned}
&\max\left(\left|n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2\right|, \left|\frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\right|\right) \\
&= n\max\left(\left|\gamma\Big(\frac{\tau}{\sqrt{n}}\Big)\right|, \left|\frac{\mu_3}{6s^3}\Big(\frac{i\tau}{\sqrt{n}}\Big)^3\right|\right) \le \frac{1}{4}\tau^2.
\end{aligned}$$

Turn to the integrand in (V.98). We then for $|\tau| < \omega\sqrt{n}$

$$\begin{aligned}
&\Big|\exp\Big\{n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2\Big\} - 1 - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\Big| \\
&\leq \Big(\Big|n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2 - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\Big| + \frac{1}{2}\Big|\frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\Big|^2\Big) \\
&\qquad \times \exp\Big[\max\Big(\Big|n\log\varphi^{a_n}(\tau/\sqrt{n}) + \frac{1}{2}\tau^2\Big|, \Big|\frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\Big|\Big)\Big] \\
&\leq \Big(\frac{\rho|\tau|^3}{\sqrt{n}} + \frac{1}{2}\Big|\frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3\Big|^2\Big)\exp\Big(\frac{\tau^2}{4}\Big) \\
&= \Big(\frac{\rho|\tau|^3}{\sqrt{n}} + \frac{\mu_3^2\tau^6}{72ns^6}\Big)\exp\Big(\frac{\tau^2}{4}\Big).
\end{aligned}$$

Use this upper bound to obtain

$$\begin{aligned}
&\int_{|\tau|\leq\omega\sqrt{n}}\Big|\Big(\varphi^{a_n}(\tau/\sqrt{n})\Big)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3e^{-\frac{1}{2}\tau^2}\Big|d\tau \\
&\leq \int_{|\tau|\leq\omega\sqrt{n}}\exp\Big(-\frac{\tau^2}{4}\Big)\Big(\frac{\rho|\tau|^3}{\sqrt{n}} + \frac{\mu_3^2\tau^6}{72ns^6}\Big)d\tau \\
&= \frac{\rho}{\sqrt{n}}\int_{|\tau|\leq\omega\sqrt{n}}\exp\Big(-\frac{\tau^2}{4}\Big)|\tau|^3d\tau + \frac{\mu_3^2}{72ns^6}\int_{|\tau|\leq\omega\sqrt{n}}\exp\Big(-\frac{\tau^2}{4}\Big)\tau^6d\tau,
\end{aligned}$$

where both the first integral and the second integral are finite, and ρ is arbitrarily small; use Corollary 4, to obtain

$$\int_{|\tau|\leq\omega\sqrt{n}}\Big|\Big(\varphi^{a_n}(\tau/\sqrt{n})\Big)^n - e^{-\frac{1}{2}\tau^2} - \frac{\mu_3}{6\sqrt{n}s^3}(i\tau)^3e^{-\frac{1}{2}\tau^2}\Big|d\tau = o\Big(\frac{1}{\sqrt{n}}\Big).$$

This gives (V.96), and therefore we obtain

$$\Big|\bar{\pi}_n^{a_n}(x) - \phi(x) - \frac{\mu_3}{6\sqrt{n}s^3}\big(x^3 - 3x\big)\phi(x)\Big| = o\Big(\frac{1}{\sqrt{n}}\Big),$$

which concludes the proof.

2.6.2 Proof of Theorem 10

It is well known and easily checked that the conditional density $p(X_1^k = y_1^k | S_1^n = na_n)$ is invariant under any i.i.d sampling scheme in the family of densities π^α as α belongs to $Im(X_1)$

(commonly called tilting change of measure). Namely

$$p(X_1^k = y_1^k | S_1^n = na_n) = \pi^\alpha(X_1^k = y_1^k | S_1^n = na_n)$$

where on the LHS the X_i's are sampled i.i.d. under p and on the RHS they are sampled i.i.d. under π^α.

Using Bayes formula, it thus holds

$$\begin{aligned} &p(X_1 = y_1 | S_1^n = na_n) = \pi^m(X_1 = y_1 | S_1^n = na_n) \\ &= \pi^m(X_1 = y_1) \frac{\pi^m(S_2^n = na_n - y_1)}{\pi^m(S_1^n = na_n)} \\ &= \frac{\sqrt{n}}{\sqrt{n-1}} \pi^m(X_1 = y_1) \frac{\widetilde{\pi_{n-1}}(\frac{m-y_1}{s\sqrt{n-1}})}{\widetilde{\pi_n}(0)}, \end{aligned} \tag{V.100}$$

where $\widetilde{\pi_{n-1}}$ is the normalized density of S_2^n under i.i.d. sampling with the density π^{a_n}; correspondingly, $\widetilde{\pi_n}$ is the normalized density of S_1^n under the same sampling. Note that a r.v. with density π^{an} has expectation m and variance s^2. Perform a third-order Edgeworth expansion of $\widetilde{\pi_{n-1}}(z)$, using Theorem 9. It follows

$$\widetilde{\pi_{n-1}}(z) = \phi(z)\Big(1 + \frac{\mu_3}{6s^3\sqrt{n-1}}(z^3 - 3z)\Big) + o\Big(\frac{1}{\sqrt{n}}\Big),$$

The approximation of $\widetilde{\pi_n}(0)$ is

$$\widetilde{\pi_n}(0) = \phi(0)\Big(1 + o\Big(\frac{1}{\sqrt{n}}\Big)\Big).$$

Hence (V.100) becomes

$$\begin{aligned} &p(X_1 = y_1 | S_1^n = na_n) \\ &= \frac{\sqrt{n}}{\sqrt{n-1}} \pi^m(X_1 = y_1) \frac{\phi(z)}{\phi(0)} \Big[1 + \frac{\mu_3}{6s^3\sqrt{n-1}}(z^3 - 3z) + o\Big(\frac{1}{\sqrt{n}}\Big)\Big] \\ &= \frac{\sqrt{2\pi n}}{\sqrt{n-1}} \pi^m(X = y_1)\phi(z)\Big(1 + R_n + o(1/\sqrt{n})\Big), \end{aligned} \tag{V.101}$$

where

$$R_n = \frac{\mu_3}{6s^3\sqrt{n-1}}(z^3 - 3z).$$

Under condition (V.78), by Corollary (4), $\mu_3/s^3 \to 0$. This yields

$$R_n = o\Big(1/\sqrt{n}\Big),$$

which gives

$$p(X_1 = y_1 | S_1^n = na_n) = \pi^m(X = y_1)\Big(1 + o(1/\sqrt{n})\Big)$$

as claimed.

2.6.3 Proof of Proposition 1

Denote

$$z_i := \frac{m_i - y_{i+1}}{s_i\sqrt{n-i-1}}$$

where

$$s_i^2 := s^2(t_i).$$

We first state a Lemma pertaining to the order of magnitude of z_i. The proof of this Lemma is in the next Subsection

Lemma 15. *Assume that $h(x) \in \mathcal{R}$. Let t_i be defined by (V.81). Assume that $a_n \to \infty$ as $n \to \infty$ and that (V.78) holds. Then as $n \to \infty$*

$$\lim_{n\to\infty} \sup_{0\le i\le k-1} z_i = 0, \qquad \text{and} \qquad \sup_{0\le i\le k-1} z_i^2 = o\left(\frac{1}{\sqrt{n}}\right).$$

We turn to the proof of Proposition 1.

It holds by Bayes formula,

$$p_{a_n}(y_1^k) = \prod_{i=0}^{k-1} p(X_{i+1} = y_{i+1} | S_{i+1}^n = na_n - s_1^i).$$

Using the invariance of the conditional distributions under the tilting it holds, for any i between 1 and $k-1$

$$p(X_{i+1} = y_{i+1} | S_{i+1}^n = na_n - S_1^i) = \frac{\sqrt{2\pi(n-i)}}{\sqrt{n-i-1}}\pi^{m_i}(X_{i+1} = y_{i+1})\phi(z_i)\Big(1 + o(1/\sqrt{n})\Big)$$
$$= \frac{\sqrt{n-i}}{\sqrt{n-i-1}}\pi^{m_i}(X_{i+1} = y_{i+1})\Big(1 - z_i^2/2 + o(z_i^2)\Big)\Big(1 + o(1/\sqrt{n})\Big)\Big),$$

where we used a Taylor expansion in the second equality. Using once more Lemma 15, under conditions (V.78), we have as $a_n \to \infty$

$$z_i^2 = o(1/\sqrt{n}).$$

Hence we get

$$p(X_{i+1} = y_{i+1} | S_{i+1}^n = na_n - s_1^i) = \frac{\sqrt{n-i}}{\sqrt{n-i-1}} \pi^{m_i}(X_{i+1} = y_{i+1}) \Big(1 + o(1/\sqrt{n})\Big),$$

which yields

$$\begin{aligned} p(X_1^k = y_1^k | S_1^n = na_n) &= \prod_{i=0}^{k-1} \left(\frac{\sqrt{n-i}}{\sqrt{n-i-1}} \pi^{m_i}(X_{i+1} = y_{i+1}) \Big(1 + o(1/\sqrt{n})\Big) \right) \\ &= \prod_{i=0}^{k-1} \pi^{m_i}(X_{i+1} = y_{i+1}) \prod_{i=0}^{k-1} \frac{\sqrt{n-i}}{\sqrt{n-i-1}} \prod_{i=0}^{k-1} \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right) \\ &= \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right) \prod_{i=0}^{k-1} \pi^{m_i}(X_{i+1} = y_{i+1}), \end{aligned}$$

The proof is completed.

2.6.4 Proof of Lemma 15

When $n \to \infty$, it holds

$$z_i \sim m_i / (s_i \sqrt{n}).$$

From Theorem 8, it holds

$$z_i \sim \frac{\psi(t_i)}{\sqrt{n\psi'(t_i)}}.$$

Since $m_i \sim m_k$ as $n \to \infty$, it holds

$$m_i \sim \psi(t_k).$$

Hence

$$\psi(t_i) \sim \psi(t_k).$$

Case 1 : if $h(x) \in R_\beta$. Hence

$$h'(x) = x^{\beta-1} l_0(x) \left(\beta + \epsilon(x)\right).$$

Set $x = \psi(t)$; we get

$$h^{'}\left(\psi(t)\right) = \psi(t)^{\beta-1} l_0\left(\psi(t)\right)\left(\beta + \epsilon\left(\psi(t)\right)\right).$$

Notice that $\psi^{'}(t) = 1/h^{'}\left(\psi(t)\right)$; we obtain

$$\frac{\psi'(t_i)}{\psi'(t_k)} = \frac{h^{'}\left(\psi(t_k)\right)}{h^{'}\left(\psi(t_i)\right)} = \frac{\left(\psi(t_k)\right)^{\beta-1} l_0\left(\psi(t_k)\right)\left(\beta + \epsilon\left(\psi(t_k)\right)\right)}{\left(\psi(t_i)\right)^{\beta-1} l_0\left(\psi(t_i)\right)\left(\beta + \epsilon\left(\psi(t_i)\right)\right)} \longrightarrow 1,$$

where we use the slowly varying propriety of l_0. Thus it holds

$$\psi'(t_i) \sim \psi'(t_k),$$

which yields

$$z_i \sim \frac{\psi(t_k)}{\sqrt{n\psi'(t_k)}}.$$

Hence we have under condition (V.78)

$$z_i^2 \sim \frac{\psi(t_k)^2}{n\psi'(t_k)} = \frac{\psi(t_k)^2}{\sqrt{n}\psi'(t_k)}\frac{1}{\sqrt{n}} = o\left(\frac{1}{\sqrt{n}}\right),$$

which implies further that $z_i \to 0$.

Case 2 : if $h(x) \in R_\infty$. It holds $m(t_k) \geq m(t_i)$ as $n \to \infty$. Since the function $t \to m(t)$ is increasing, we have

$$t_i \leq t_k.$$

The function $t \to \psi^{'}(t)$ is decreasing, since

$$\psi^{''}(t) = -\frac{\psi(t)}{t^2}\epsilon(t)\left(1 + o(1)\right) < 0 \qquad as \quad t \to \infty.$$

Therefore as $n \to \infty$

$$\psi'(t_i) \geq \psi'(t_k) > 0,$$

which yields

$$z_i \sim \frac{\psi(t_i)}{\sqrt{n\psi'(t_i)}} \leq \frac{2\psi(t_k)}{\sqrt{n\psi'(t_k)}},$$

hence we have

$$z_i^2 \leq \frac{4\psi(t_k)^2}{n\psi'(t_k)} = \frac{4\psi(t_k)^2}{\sqrt{n}\psi'(t_k)}\frac{1}{\sqrt{n}} = o\left(\frac{1}{\sqrt{n}}\right),$$

where the last step holds from condition (V.78). Further it holds $z_i \to 0$.

This closes the proof of the Lemma.

2.6.5 Proof of Lemma 12

Case 1 : if $h(t) \in R_\beta$. By Theorem 8, it holds $s^2 \sim \psi'(t)$ with $\psi(t) \sim t^{1/\beta} l_1(t)$, where l is some slowly varying function. Consider $\psi'(t) = 1/h'\big(\psi(t)\big)$, hence

$$\begin{aligned}\frac{1}{s^2} &\sim h'\big(\psi(t)\big) = \psi(t)^{\beta-1} l_0\big(\psi(t)\big)\Big(\beta + \epsilon\big(\psi(t)\big)\Big) \\ &\sim \beta t^{1-1/\beta} l_1(t)^{\beta-1} l_0\big(\psi(t)\big) = o(t),\end{aligned}$$

where $l_0 \in R_0$. This implies for any $u \in K$

$$\frac{u}{s} = o(\sqrt{t}),$$

which yields, using (V.71)

$$\begin{aligned}\frac{s^2\,(t+u/s)}{s^2} \sim \frac{\psi'(t+u/s)}{\psi'(t)} &= \frac{\psi(t)^{\beta-1} l_0\big(\psi(t)\big)\Big(\beta + \epsilon\big(\psi(t)\big)\Big)}{\Big(\psi(t+u/s)\Big)^{\beta-1} l_0\big(\psi(t+u/s)\big)\Big(\beta + \epsilon\big(\psi(t+u/s)\big)\Big)} \\ &\sim \frac{\psi(t)^{\beta-1}}{\psi(t+u/s)^{\beta-1}} \sim \frac{t^{1-1/\beta} l_1(t)^{\beta-1}}{(t+u/s)^{1-1/\beta} l_1(t+u/s)^{\beta-1}} \longrightarrow 1.\end{aligned}$$

Case 2 : if $h(t) \in R_\infty$. Then $\psi(t) \in \widetilde{R_0}$, hence it holds

$$\frac{1}{st} \sim \frac{1}{t\sqrt{\psi'(t)}} = \sqrt{\frac{1}{t\psi(t)\epsilon(t)}} \longrightarrow 0,$$

which last step holds from condition (V.74). Hence for any $u \in K$, we get as $n \to \infty$

$$\frac{u}{s} = o(t),$$

thus using the slowly varying propriety of $\psi(t)$ we have

$$\begin{aligned}\frac{s^2\,(t+u/s)}{s^2} \sim \frac{\psi'(t+u/s)}{\psi'(t)} &= \frac{\psi(t+u/s)\epsilon(t+u/s)}{t+u/s}\frac{t}{\psi(t)\epsilon(t)} \\ &\sim \frac{\epsilon(t+u/s)}{\epsilon(t)} = \frac{\epsilon(t) + O\big(\epsilon'(t)u/s\big)}{\epsilon(t)} \longrightarrow 1, \qquad \text{(V.102)}\end{aligned}$$

where we used a Taylor expansion in the second line, and where the last step holds from condition (V.74). This completes the proof.

2.6.6 Proof of Theorem 12

Making use of

$$p(X_1^k = y_1^k | S_1^n = na_n) = \prod_{i=0}^{k-1} p(X_{i+1} = y_{i+1} | S_{i+1}^n = na_n - s_1^i),$$

and using the tilted density π^{a_n} instead of π^{m_i} it holds

$$p(X_{i+1} = y_{i+1} | S_{i+1}^n = na_n - s_1^i) = \frac{\sqrt{n-i}}{\sqrt{n-i-1}} \pi^{a_n}(X_{i+1} = y_{i+1}) \frac{\widetilde{\pi_{n-i-1}}\left(\frac{(i+1)a_n - s_1^{i+1}}{s\sqrt{n-i-1}}\right)}{\widetilde{\pi_{n-i}}\left(\frac{ia_n - s_1^i}{s\sqrt{n-i}}\right)}, \quad \text{(V.103)}$$

where $\widetilde{\pi_{n-i-1}}$ is the normalized density of S_{i+2}^n under i.i.d. sampling with π^{a_n}. Correspondingly, denote $\widetilde{\pi_{n-i}}$ the normalized density of S_{i+1}^n under the same sampling. Write

$$z_i = \frac{ia_n - s_1^{i-1}}{s\sqrt{n-i+1}}.$$

By Theorem 9 a third-order Edgeworth expansion yields

$$\widetilde{\pi_{n-i-1}}(z_i) = \phi(z_i)\left(1 + R_n^i\right) + o\left(\frac{1}{\sqrt{n}}\right),$$

where

$$R_n^i = \frac{\mu_3}{6s^3\sqrt{n-i-1}}(z_i^3 - 3z_i).$$

Accordingly

$$\widetilde{\pi_{n-i}}(z_{i-1}) = \phi(z_{i-1})\left(1 + R_n^{i-1}\right) + o\left(\frac{1}{\sqrt{n}}\right).$$

When $a_n \to \infty$, using Theorem 8, it holds

$$\begin{aligned} \sup_{0 \le i \le k-1} z_i^2 \sim \frac{(i+1)^2 a_n^2}{s^2 n} &\le \frac{2k^2 a_n^2}{s^2 n} = \frac{2k^2 (m(t))^2}{s^2 n} \\ &\sim \frac{2k^2(\psi(t))^2}{\psi'(t)n} = \frac{2k^2(\psi(t))^2}{\sqrt{n}\psi'(t)} \frac{1}{\sqrt{n}} = o\left(\frac{1}{\sqrt{n}}\right), \end{aligned} \quad \text{(V.104)}$$

where the last step holds under condition (V.78). Hence it holds $z_i \to 0$ for $0 \leq i \leq k-1$ as $a_n \to \infty$, and by Corollary 4, $\mu_3/s^3 \to 0$; Hence

$$R_n^i = o\left(1/\sqrt{n}\right) \text{ and } R_n^{i-1} = o\left(1/\sqrt{n}\right).$$

We thus get

$$\begin{aligned} p(X_{i+1} = y_{i+1}|S_{i+1}^n = na_n - s_1^i) &= \frac{\sqrt{n-i}}{\sqrt{n-i-1}}\pi^{a_n}(X_{i+1} = y_{i+1})\frac{\phi(z_i)}{\phi(z_{i-1})}\left(1 + o(1/\sqrt{n})\right) \\ &= \frac{\sqrt{n-i}}{\sqrt{n-i-1}}\pi^{a_n}(X_{i+1} = y_{i+1})\left(1 - (z_i^2 - z_{i-1}^2)/2 + o(z_i^2 - z_{i-1}^2)\right)\left(1 + o(1/\sqrt{n})\right), \end{aligned}$$

where we used a Taylor expansion in the second equality. Using (V.104), we have as $a_n \to \infty$

$$|z_i^2 - z_{i-1}^2| = o(1/\sqrt{n}),$$

from which

$$p(X_{i+1} = y_{i+1}|S_{i+1}^n = na_n - s_1^i) = \frac{\sqrt{n-i}}{\sqrt{n-i-1}}\pi^{a_n}(X_{i+1} = y_{i+1})\left(1 + o(1/\sqrt{n})\right),$$

which yields

$$\begin{aligned} p(X_1^k = y_1^k|S_1^n = na_n) &= \prod_{i=0}^{k-1}\left(\pi^{a_n}(X_{i+1} = y_{i+1})\sqrt{\frac{n}{n-k}}\right)\prod_{i=0}^{k-1}\left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right) \\ &= \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right)\prod_{i=0}^{k-1}\pi^{a_n}(X_{i+1} = y_{i+1}). \end{aligned}$$

This completes the proof.

2.6.7 Proof of Lemma 13

For a density $p(x)$ defined in as in (V.67), we show that $g(x)$ is a convex function when x is large. If $h(x) \in R_\beta$, for x large

$$g''(x) = h'(x) = \frac{h(x)}{x}\left(\beta + \epsilon(x)\right) > 0.$$

If $h(x) \in R_\infty$, its reciprocal function $\psi(x) \in \widetilde{R_0}$. Set $x := \psi(v)$. Then

$$g''(x) = h'(x) = \frac{1}{\psi'(v)} = \frac{v}{\psi(v)\epsilon(v)} > 0,$$

where the inequality holds since $\epsilon(v) > 0$ when v is large enough. Hence $g(x)$ is convex for large x. Therefore, the density $p(x)$ with $h(x) \in \mathcal{R}$ satisfies the conditions of Theorem 6.2.1 in (Jensen, 1995). Denote by p_n the density of $\bar{X} = (X_1 + \ldots + X_n)/n$. We obtain from formula (2.2.6) of (Jensen, 1995), using a third order Edgeworth expansion

$$P(S_1^n \geq na_n) = \frac{\Phi(t)^n \exp(-nta_n)}{\sqrt{n}ts(t)} \left(B_0(\lambda_n)\right) + O\left(\frac{\mu_3(t)}{6\sqrt{n}s^3(t)} B_3(\lambda_n)\right),$$

where $\lambda_n = \sqrt{n}ts(t)$, $B_0(\lambda_n)$ and $B_3(\lambda_n)$ are defined by

$$B_0(\lambda_n) = \frac{1}{\sqrt{2\pi}}\left(1 - \frac{1}{\lambda_n^2} + o(\frac{1}{\lambda_n^2})\right), \qquad B_3(\lambda_n) \sim -\frac{3}{\sqrt{2\pi}\lambda_n}.$$

We show that as $a_n \to \infty$

$$\frac{1}{\lambda_n^2} = o\left(\frac{1}{n}\right). \tag{V.105}$$

Since $n/\lambda_n^2 = 1/(t^2 s^2(t))$, (V.105) is equivalent to show that

$$t^2 s^2(t) \longrightarrow \infty.$$

By Theorem 8, $m(t) \sim \psi(t)$ and $s^2(t) \sim \psi'(t)$; combined with $m(t) = a_n$, it holds $t \sim h(a_n)l_1(a_n)$, where l_1 is some slowly varying function. If $h \in R_\beta$, notice that

$$\psi'(t) = \frac{1}{h'(\psi(t))} = \frac{\psi(t)}{h\left(\psi(t)\right)\left(\beta + \epsilon(\psi(t))\right)} \sim \frac{a_n}{h(a_n)\left(\beta + \epsilon(\psi(t))\right)};$$

hence

$$t^2 s^2(t) \sim h(a_n)^2 l_1(a_n)^2 \frac{a_n}{h(a_n)\left(\beta + \epsilon(\psi(t))\right)} = \frac{a_n h(a_n) l_1(a_n)^2}{\beta + \epsilon(\psi(t_n))} \longrightarrow \infty.$$

If $h \in R_\infty$, then $\psi(t) \in \widetilde{R_0}$, thus

$$t^2 s^2(t) \sim t^2 \frac{\psi(t)\epsilon(t)}{t} = t\psi(t)\epsilon(t) \longrightarrow \infty,$$

Summing up we have proved that

$$B_0(\lambda_n) = \frac{1}{\sqrt{2\pi}}\left(1 + o\left(\frac{1}{n}\right)\right).$$

By (V.105), λ_n goes to ∞ as $a_n \to \infty$; this implies further that $B_3(\lambda_n) \to 0$. On the other hand, by Corollary 4 it holds $\mu_3/s^3 \to 0$. Hence we obtain

$$P(S_1^n \geq na_n) = \frac{\Phi(t)^n \exp(-nta_n)}{\sqrt{2\pi n}ts(t)}\left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right),$$

which gives (V.86). By Jensen's Theorem 6.2.1 (Jensen, 1995) and formula (2.2.4) in (Jensen, 1995) it follows uniformly in τ

$$p(S_1^n/n = \tau) = \frac{\sqrt{n}\Phi(t_\tau)^n \exp(-nt_\tau\tau)}{\sqrt{2\pi}s(t_\tau)}\left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right),$$

which, together with $p(S_1^n = n\tau) = (1/n)p(S_1^n/n = \tau)$, gives (V.87).

2.6.8 Proof of Theorem 17

It holds

$$\begin{aligned}
p_{A_n}(y_1) &= \int_{a_n}^{\infty} p(X_1 = y_1|S_1^n = n\tau)p(S_1^n = n\tau|S_1^n \geq na_n)d\tau \\
&= \frac{p(X_1 = y_1)}{P(S_1^n \geq na_n)}\int_{a_n}^{\infty} p(S_2^n = n\tau - y_1)d\tau \\
&= \left(1 + \frac{P_2}{P_1}\right)\frac{p(X_1 = y_1)}{P(S_1^n \geq na_n)}\int_{a_n}^{a_n+\eta_n} p(S_2^n = n\tau - y_1)d\tau \\
&= \left(1 + \frac{P_2}{P_1}\right)\int_{a_n}^{a_n+\eta_n} p(X_1 = y_1|S_1^n = n\tau)p(S_1^n = n\tau|S_1^n \geq na_n)d\tau \qquad \text{(V.106)}
\end{aligned}$$

where the second equality is obtained by Bayes formula, and

$$P_1 = \int_{a_n}^{a_n+\eta_n} p(S_2^n = n\tau - y_1)d\tau,$$

$$P_2 = \int_{a_n+\eta_n}^{\infty} p(S_2^n = n\tau - y_1)d\tau.$$

We show that P_2 is infinitely small with respect to P_1. Indeed

$$P_2 = \frac{1}{n} P\left(S_2^n \geq n(a_n + \eta_n) - y_1\right) = \frac{1}{n} P\left(S_2^n \geq (n-1)c_n\right),$$
$$P_1 + P_2 = \frac{1}{n} P\left(S_2^n \geq na_n - y_1\right) = \frac{1}{n} P\left(S_2^n \geq (n-1)d_n\right),$$

where $c_n = \left(n(a_n + \eta_n) - y_1\right)/(n-1)$ and $d_n = (na_n - y_1)/(n-1)$. Denote $t_{c_n} = m^{-1}(c_n)$ and $t_{d_n} = m^{-1}(d_n)$. Using Lemma 13, it holds

$$\frac{P_2}{P_1 + P_2} = \left(+o\left(\frac{1}{\sqrt{n}}\right)\right) \frac{t_{d_n} s(t_{d_n})}{t_{c_n} s(t_{c_n})} \exp\left(-(n-1)\left(I(c_n) - I(d_n)\right)\right).$$

Using the convexity of the function I, it holds

$$\begin{aligned}\exp\left(-(n-1)I(c_n) - I(d_n)\right) &\leq \exp -(n-1)(c_n - d_n)m^{-1}(d_n) \\ &= \exp -n\eta_n m^{-1}(d_n).\end{aligned}$$

The function $u \to m^{-1}(u)$ is increasing. Since $d_n \geq a_n$ as $a_n \to \infty$, it holds $m^{-1}(d_n) \geq m^{-1}(a_n)$; hence $\exp -(n-1)\left(I(c_n) - I(d_n)\right) \leq \exp -n\eta_n m^{-1}(a_n) \longrightarrow 0$. We now show that

$$\frac{t_{d_n} s(t_{d_n})}{t_{c_n} s(t_{c_n})} \longrightarrow 1.$$

By definition, $c_n/d_n \to 1$ as $a_n \to \infty$. If $h \in R_\beta$, it holds

$$\left(\frac{t_{d_n} s(t_{d_n})}{t_{c_n} s(t_{c_n})}\right)^2 \sim \left(\frac{d_n h(d_n)}{\beta + \epsilon\left(\psi(d_n)\right)}\right)^2 \left(\frac{\beta + \epsilon\left(\psi(c_n)\right)}{c_n h(c_n)}\right)^2 \sim \left(\frac{h(d_n)}{h(c_n)}\right)^2 \longrightarrow 1.$$

If $h \in R_\infty$,

$$t^2 s^2(t) \sim t\psi(t)\epsilon(t),$$

hence

$$\left(\frac{t_{d_n} s(t_{d_n})}{t_{c_n} s(t_{c_n})}\right)^2 \sim \frac{d_n \psi(d_n)\epsilon(d_n)}{c_n \psi(c_n)\epsilon(c_n)} \sim \frac{\epsilon(d_n)}{\epsilon(c_n)} = \frac{\epsilon\left(c_n - n\eta_n/(n-1)\right)}{\epsilon(c_n)} \longrightarrow 1,$$

where last step holds by using the same argument as in the second line of (V.102). We obtain

$$\frac{P_2}{P_1} = o\left(1\right).$$

Therefore $p_{A_n}(y_1)$ can be approximated by

$$p_{A_n}(y_1) = (1 + o(1)) \int_{a_n}^{a_n+\eta_n} p(X_1 = y_1 | S_1^n = n\tau) p(S_1^n = n\tau | S_1^n \geq na_n) d\tau.$$

By Lemma 13, it follows that uniformly when $\tau \in [a_n, a_n + \eta_n]$

$$\begin{aligned} p(S_1^n = n\tau | S_1^n \geq na_n) &= \frac{p(S_1^n = n\tau)}{P(S_1^n \geq na_n)} \\ &= \left(1 + o\left(\frac{1}{\sqrt{n}}\right)\right) \frac{ts(t)}{s(t_\tau)} \exp\left(-n(I(\tau) - I(a_n))\right), \end{aligned} \tag{V.107}$$

We now turn back to (V.106) and note that under the appropriate condition (V.78) or (V.79) the corresponding approximating density π^τ or g_τ can be seen to hold uniformly on τ in $(a_n, a_n+\eta_n)$. Inserting (V.107) into (V.106), we complete the proof of Theorem 17 insering the corresponding local result.

These two articles represent theoretical basis of a potential method for solving inverse problems using extreme deviations. A lot of demonstrations are still required before applying this method to a concrete case.

Conclusion générale et perspectives

Les apports scientifiques A travers le travail mené durant cette thèse, nous avons pu traiter dans sa globalité un problème d'optimisation robuste sous contraintes pour un code de calculs que nous voyons comme une boîte blanche.
La première étape consiste à diminuer la dimension du problème par des méthodes de réduction de la dimension. Ensuite, si le nombre de contraintes n'est pas trop important, il est possible d'établir des méta-modèles pour la sortie d'intérêt ainsi que pour chacune des contraintes. Ceci permet, lors des études d'optimisation par exemple, de ne pas faire appel au code de calculs mais de se servir uniquement des méta-modèles, ce qui réduit considérablement les temps de calculs. Les principales méthodes de réduction de la dimension et de méta-modélisation ont été décrites dans ce mémoire.
L'étape suivante consiste à poser le problème d'optimisation. Ici, nous voulions obtenir un optimum robuste sous contraintes. Dans ce cas, il est essentiel de recenser et de modéliser les différentes incertitudes du système. Puis il faut propager ces incertitudes à travers le modèle afin d'obtenir une estimation des moments pour la sortie d'intérêt. Ce sont ces moments qui représentent les objectifs de l'optimisation robuste, définie comme une optimisation bi-objectifs. Plusieurs méthodes permettent de résoudre de tels problèmes, sous contraintes. Elles ont été décrites ainsi que l'ensemble de la méthodologie de conception robuste.
A l'issue de cette optimisation, il se peut que l'optimum que nous obtenons ne satisfasse pas une ou plusieurs contrainte(s) alors que l'optimum est intéressant et présente d'excellentes qualités de robustesse. Dans ce cas, il peut être judicieux de faire appel à des méthodes de résolution de problèmes inverses. Le but est de trouver des combinaisons en entrée qui satisfont la ou les contrainte(s) tout en restant proche de l'optimum trouvé. Cette dernière condition devient alors une nouvelle contrainte. Pour cela, plusieurs méthodes existent dans la littérature. Décrites dans ce mémoire, ces méthodes présentent certains inconvénients que nous avons présentés. Il

s'agit notamment de l'obligation de régulariser le problème pour les méthodes usuelles, qui ne permettent de trouver qu'une seule solution. D'autres méthodes sont très coûteuses, certaines sont complexes. Face à ces inconvénients et dans le but de pouvoir obtenir plusieurs solutions du problème inverse, nous avons développé deux nouvelles méthodes, MRM et COMET. Une troisième méthode, qui est encore en développement, est également exposée dans ce mémoire. Basée sur la théorie des valeurs extrêmes, elle fait l'objet de deux articles traitant des aspects théoriques constituant les bases de la méthode.
Les deux premières méthodes s'appliquent en toutes dimensions et permettent d'obtenir beaucoup de solutions dans une tolérance choisie autour de la valeur cible. Ces solutions sont en plus bien réparties. La méthode MRM a l'inconvénient de n'être applicable que sur des fonctions monotones. Au contraire, la méthode COMET ne nécessite aucune hypothèse forte sur la fonction étudiée. Ceci permet de l'appliquer directement sur le code de calculs et ainsi, de prendre en compte les contraintes du système. Ces méthodes ont également été testées sur différentes fonctions usuelles et dans différentes dimensions. Ceci a permis de mettre en évidence les possibilités de la méthode COMET.

Les apports industriels Ces trois étapes ont été menées sur un cas test industriel dont l'objectif était d'améliorer et d'accélérer les études de dimensionnement en avant-projets. L'amélioration des résultats en avant-projets consiste à réaliser une optimisation robuste à la place d'une optimisation globale dont l'optimum peut ne pas être robuste à des variations sur les entrées. L'accélération des études passe par la réduction des rebouclages entre les différents métiers lors du dimensionnement du moteur. Avec l'exemple du compresseur HP, nous avons pu tester les différentes méthodes décrites dans ce mémoire. Dans un premier temps, nous effectuons la réduction de la dimension et la méta-modélisation pour les sorties d'intérêt. Ensuite, nous avons déterminé l'optimum robuste pour la masse du CHP. L'optimum trouvé ne satisfaisant pas une des contraintes du système, on parle d'un problème d'intégration. Nous avons alors utilisé les méthodes de résolution d'un problème inverse développées durant cette thèse. La méthode COMET, appliquée directement sur le code de calculs, permet d'obtenir plusieurs solutions possibles satisfaisant toutes les contraintes. Ces résultats sont très prometteurs, ce qui conduit Safran Aircraft Engine à perpétuer l'étude et la mise en œuvre de ces méthodes.
Enfin, nous avons étudié un second cas test industriel, portant sur l'étude des efforts sur les supports-paliers. Une première partie consistait à établir un méta-modèle permettant d'exprimer assez fidèlement les déformations subies par la pièce par rapport à l'effort qui lui est appliqué. Cet effort est caractérisé par une intensité et une direction. L'établissement du méta-modèle montre les limites des modèles usuels qui ne peuvent pas prendre en compte les carac-

téristiques physiques du phénomène. Dans ce cas, nous devons construire notre propre modèle paramétrique. La seconde étape consistait à utiliser ce modèle pour retrouver, lors d'un essai, l'effort appliqué sur la pièce à partir des déformations mesurées. Il s'agit d'un problème d'inversion avec une unique solution. L'application des méthodes développées dans cette thèse a montré qu'il était suffisant d'utiliser des méthodes usuelles de type Newton pour trouver une solution.

Les perspectives scientifiques Dans cette thèse, la partie sur la conception robuste n'a consisté qu'à décrire la méthodologie et à exposer les différentes méthodes existantes. Ces méthodes sont très coûteuses en temps de calculs et, parfois, ne permettent pas de converger vers l'optimum puisqu'elles atteignent d'abord un nombre maximal d'itérations. Il serait intéressant de travailler sur la formulation du problème et les méthodes de propagation afin d'estimer efficacement les moments de la sortie, avec le moins d'appels possible au code de calculs. Parmi les méthodes d'optimisation, nous avons évoqué la méthode de l'entropie croisée dont les résultats en optimisation globale sont très bons. Il pourrait être intéressant de généraliser cette méthode à l'optimisation multi-objectifs afin de la comparer aux méthodes existantes.
Les principaux développements de cette thèse ont porté sur la résolution de problèmes inverses. La méthode la plus prometteuse est la méthode COMET, qui a fait et fera l'objet de plusieurs articles scientifiques et industriels. Il s'agira d'étudier le dernier algorithme présenté, celui avec les chaînes. En effet, il est possible de tirer de nombreuses informations des chaînes construites, puisqu'une structure de liens est établie. Nous pourrons notamment nous intéresser à la vitesse de convergence de la méthode et aux façons de l'améliorer.
La méthode basée sur les grandes déviations fera également l'objet de futurs développements. En effet, de nombreux travaux sont encore à effectuer afin d'obtenir une méthode applicable en pratique.
Les perspectives industrielles Les applications de la méthode de contours pourront être nombreuses à Safran Aircraft Engine. Nous envisageons de la tester sur des problèmes d'estimation des paramètres d'une loi par maximum de vraisemblance. En effet, ces problèmes consistent à résoudre des systèmes où chaque équation est l'annulation d'une dérivée partielle de la vraisemblance. La méthode COMET permettrait de résoudre des cas difficiles de mélanges de lois conduisant à des systèmes où le nombre d'inconnues est supérieur au nombre d'équations.
Nous pourrions également l'appliquer sur des problèmes utilisant des méthodes inverses usuelles pour lesquelles le problème a dû être régularisé. Plus généralement, elles peuvent s'appliquer à tout problème où on veut atteindre une cible en sortie d'un calcul et on souhaite trouver les

ensembles solutions en entrée. Pour cela, il serait bon de pousser plus loin les tests sur cette méthode. Il s'agirait notamment de la tester en très grandes dimensions et sur des sorties discrètes.

Une autre perspective de la méthode COMET consiste à prendre en compte les incertitudes des entrées afin de fournir des solutions avec leur incertitude.

Le but ultime de l'utilisation de cette méthode à Safran Aircraft Engine est de l'implémenter dans les outils de la société afin de l'utiliser de manière interactive. Pour cela, une optimisation du code par une parallélisation des calculs permettrait d'accélérer la recherche des solutions. Par exemple, on peut diviser le domaine et rechercher les solutions sur chaque sous-domaine simultanément. Une étape importante serait également d'utiliser un logiciel de programmation assurant des appels rapides aux outils effectuant les calculs.

Bibliographie

[1] Aleksander, I. and H. Morton (1990). *An introduction to neural computing*, Volume 3. Chapman & Hall London.

[2] Altınel, I. K., N. Aras, and K. C. Özkısacık (2011). Variable neighbourhood search heuristics for the probabilistic multi-source weber problem. *Journal of the Operational Research Society 62*(10), 1813–1826.

[3] Angel-Bello, F. R., J. L. González-Velarde, and A. M. Alvarez (2006). Greedy randomized adaptive search procedures. In *Metaheuristic Procedures for Training Neutral Networks*, pp. 207–223. Springer.

[4] Au, S. K. and J. L. Beck (1999). A new adaptive importance sampling scheme for reliability calculations. *Structural safety 21*(2), 135–158.

[5] Bäck, T. (1996). *Evolutionary algorithms in theory and practice : evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

[6] Balkema, A. A., C. Klüppelberg, and S. I. Resnick (1993). Densities with gaussian tails. *Proc. London Math. Soc 66*(3), 568–588.

[7] Bar-Lev, S. K., D. Bshouty, and P. Enis (1992). On polynomial variance functions. *Probability theory and related fields 94*(1), 69–82.

[8] Barndorff-Nielsen, O. (1978). *Information and exponential families in statistical theory*. Chichester : John Wiley and Sons.

[9] Barton, R. R. (1992). Metamodels for simulation input-output relations. In *Proceedings of the 24th conference on Winter simulation*, pp. 289–299. ACM.

[10] Barton, R. R. (1994). Metamodeling : a state of the art review. In *Proceedings of the 26th conference on Winter simulation*, pp. 237–244. Society for Computer Simulation International.

[11] Benoist, D., Y. Tourbier, and S. German-Tourbier (1994). *Plan d'expériences : construction et analyse*. Edition Lavoisier Tec & Doc.

[12] Beyer, H.-G. and H.-P. Schwefel (2002). Evolution strategies–a comprehensive introduction. *Natural computing 1*(1), 3–52.

[13] Beyer, H.-G. and B. Sendhoff (2007). Robust Optimization - A Comprehensive Survey. *Computer Methods in Applied Mechanics and Engineering*.

[14] Bingham, N. H., C. M. Goldie, and J. L. Teugels (1987). *Regular variation*. Cambridge University Press.

[15] Biret, M., M. Achibi, and M. Broniatowski (2014). Recherche des ensembles de niveaux d'une fonction multi variée à valeurs réelles sous conditions de monotonie. *I-Revues CNRS, Actes du Congrès Lambda-Mu 19*.

[16] Biret, M. and M. Broniatowski (2016). Safip : a streaming algorithm for inverse problems. *arXiv preprint arXiv :1609.08328*.

[17] Biret, M., M. Broniatowski, and Z. Cao (2015). A sharp abelian theorem for the laplace transform. In *Mathematical Statistics and Limit Theorems*, pp. 67–92. Springer.

[18] Biret, M., M. Broniatowski, and Z. Cao (2016). A gibbs conditional theorem under extreme deviation. *arXiv preprint arXiv :1610.04052*.

[19] Björck, A. (1990). *Numerical methods for least squares problems*. In Handbook of Numerical Analysis, P. G. Ciarlet and J. L.Lions, eds., Elsevier.

[20] Blum, C. and A. Roli (2003). Metaheuristics in combinatorial optimization : Overview and conceptual comparison. *ACM Computing Surveys (CSUR) 35*(3), 268–308.

[21] Bonabeau, E., M. Dorigo, and G. Theraulaz (1999). *Swarm intelligence : from natural to artificial systems*. Number 1. Oxford university press.

[22] Borovkov, A. (2008). Tauberian and abelian theorems for rapidly decresing distributions and their applications to stable laws. *Siberian Mathematical Journal 49*(5), 796–805.

[23] Bousquet, N. (2012). Accelerated monte carlo estimation of exceedance probabilities under monotonicity constraints. *Annales de la Faculté des Sciences de Toulouse. Série 6 21*, 557–591.

[24] Box, G. E. P., W. G. Hunter, J. S. Hunter, et al. (1978). Statistics for experimenters.

[25] Brimberg, J., P. Hansen, and N. Mladenovic (2010). Attraction probabilities in variable neighborhood search. *4OR 8*(2), 181–194.

[26] Broniatowsk, M. and D. M. Mason (1994). Extended large deviations. *Journal of Theoretical Probability 7*(3), 647–666.

[27] Broniatowski, M. and Z. Cao (2012). Stretched random walks and the behaviour of their summands. *arXiv preprint arXiv :1205.5936*.

[28] Broniatowski, M., V. Caron, et al. (2014). Long runs under a conditional limit distribution. *The Annals of Applied Probability 24*(6), 2246–2296.

[29] Broniatowski, M. and G. Celant (2014). Some overview on unbiased interpolation and extrapolation designs. *arXiv preprint arXiv :1403.5113*.

[30] Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 577–593.

[31] Burba, F., F. Ferraty, and P. Vieu (2008). Convergence de l'estimateur à noyau des k plus proches voisins en régression fonctionnelle non-paramétrique. *Comptes Rendus Mathematiques 346*(5), 339–342.

[32] Coello, C. A. C., D. A. V. Veldhuizen, and G. B. Lamont (2002). *Evolutionary algorithms for solving multi-objective problems*, Volume 242. Springer.

[33] Collette, Y. and P. Siarry (2002). *Optimisation multiobjectif*. Editions Eyrolles.

[34] Csiszár, I. (1984). Sanov property, generalized i-projection and a conditional limit theorem. *The Annals of Probability*, 768–793.

[35] Cukier, R. I., H. B. Levin, and K. E. Shuler (1978). Nonlinear sensitivity analysis of multiparameter model systems. *Journal of computational physics 26*(1), 1–42.

[36] Cunningham, P. (2008). Dimension reduction. In *Machine learning techniques for multimedia*, pp. 91–112. Springer.

[37] Das, I. and J. E. Dennis (1998). Normal-boundary intersection : A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization 8*(3), 631–657.

[38] de Oliveira, L. S. and S. F. P. Saramago (2010). Multiobjective optimization techniques applied to engineering problems. *Journal of the brazilian society of mechanical sciences and engineering 32*(1), 94–105.

[39] de Rocquigny, E., N. Devictor, and S. Tarantola (2008). *Uncertainty in Industrial Practice : A Guide to Quantitative Uncertainty Management*. Wiley.

[40] Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). A fast and elitist multiobjective genetic algorithm : Nsga-ii. *Evolutionary Computation, IEEE Transactions on 6*(2), 182–197.

[41] Dembo, A. and O. Zeitouni (1996). Refinements of the gibbs conditioning principle. *Probability theory and related fields 104*(1), 1–14.

[42] Diaconis, P. and D. A. Freedman (1988). Conditional limit theorems for exponential families and finite versions of de finetti's theorem. *Journal of Theoretical Probability 1*(4), 381–410.

[43] Ditlevsen, O., R. Olesen, and G. Mohr (1986). Solution of a class of load combination problems by directional simulation. *Structural Safety 4*(2), 95–109.

[44] Dréo, J., A. Pétrowski, É. D. Taillard, and P. Siarry (2003). Métaheuristiques pour l'optimisation difficile. *Eyrolles (Editions)*.

[45] Dubois, D., M. A. Lubiano, H. Prade, M. A. Gil, P. Grzegorzewski, and O. Hryniewicz (2008). *Soft methods for handling variability and imprecision*, Volume 48. Springer Science & Business Media.

[46] Eberhart, R. C., Y. Shi, and J. Kennedy (2001). *Swarm intelligence*. San Mateo, CA : Morgan Kaufmann, Elsevier.

[47] Engl, H. W., M. Hanke, and A. Neubauer (1996). *Regularization of inverse problems*, Volume 375. Springer Science & Business Media.

[48] Fang, K.-T., D. K. Lin, P. Winker, and Y. Zhang (2000). Uniform design : theory and application. *Technometrics 42*(3), 237–248.

[49] Feigin, P. D. and E. Yashchin (1983). On a strong tauberian result. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 65*(1), 35–48.

[50] Feller, W. (1971). *An introduction to probability theory and its applications* (Second ed.), Volume II. John Wiley and Sons.

[51] Fodor, I. K. (2002). A survey of dimension reduction techniques. *Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory*.

[52] Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, 1–67.

[53] Friedman, J. H. (2001). Greedy function approximation : a gradient boosting machine. *Annals of statistics*, 1189–1232.

[54] Friedman, J. H. and W. Stuetzle (1981). Projection pursuit regression. *Journal of the American statistical Association 76*(376), 817–823.

[55] Furnival, G. M. and R. W. Wilson (1974). Regressions by leaps and bounds. *Technometrics 16*(4), 499–511.

[56] Gelfand, A. E. and A. F. M. Smith (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association 85*(410), 398–409.

[57] Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences 8*(1), 156–166.

[58] Glover, F. and G. A. Kochenberger (2003). *Handbook of metaheuristics*. Springer Science & Business Media.

[59] Goldberg, D. E., V. Corruble, J.-G. Ganascia, and J. Holland (1994). *Algorithmes génétiques : exploration, optimisation et apprentissage automatique*. Addison-Wesley France.

[60] Golub, G. H. and C. F. V. Loan (2012). *Matrix computations*, Volume 3. JHU Press.

[61] Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In *ICANN 98*, pp. 201–206. Springer.

[62] Gu, C. (2013). *Smoothing spline ANOVA models*, Volume 297. Springer Science & Business Media.

[63] Gunn, S. R. et al. (1998). Support vector machines for classification and regression. *ISIS technical report 14*.

[64] György, A. and L. Kocsis (2011). Efficient multi-start strategies for local search algorithms. *Journal of Artificial Intelligence Research*, 407–444.

[65] Haldar, A. and S. Mahadevan (2000). *Probability, reliability, and statistical methods in engineering design*, Volume 1. Wiley New York.

[66] Hartigan, J. A. (1975). *Clustering algorithms*. John Wiley & Sons, Inc.

[67] Hastie, T. J. and R. J. Tibshirani (1990). *Generalized additive models*, Volume 43. CRC Press.

[68] Hedayat, A. S., N. J. A. Sloane, and J. Stufken (2012). *Orthogonal arrays : theory and applications*. Springer Science & Business Media.

[69] Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1998). Bayesian model averaging. In *In Proceedings of the AAAI Workshop on Integrating Multiple Learned Models*, pp. 77–83. Citeseer.

[70] Homma, T. and A. Saltelli (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety 52*(1), 1–17.

[71] Iman, R. and W. Conover (1979). The use of the rank transformation in regression. *Technometrics 21*(4), 4997.

[72] Iman, R. L. and J. C. Helton (1991). The repeatability of uncertainty and sensitivity analyses for complex probabilistic risk assessments. *Risk Analysis 11*(4), 591–606.

[73] Jeffreys, H. and B. Jeffreys (1999). *Methods of mathematical physics*. Cambridge university press.

[74] Jensen, J. L. (1995). *Saddlepoint approximations*. Number 16. Oxford University Press.

[75] Jin, R., W. Chen, and T. W. Simpson (2001). Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization 23*(1), 1–13.

[76] Jin, R., W. Chen, and A. Sudjianto (2002). On sequential sampling for global metamodeling in engineering design. In *ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 539–548. American Society of Mechanical Engineers.

[77] Johnson, M. E., L. M. Moore, and D. Ylvisaker (1990). Minimax and maximin distance designs. *Journal of statistical planning and inference 26*(2), 131–148.

[78] Jørgensen, B., J. R. Martínez, et al. (1997). Tauber theory for infinitely divisible variance functions. *Bernoulli 3*(2), 213–224.

[79] Juszczak, D. and A. Nagaev (2004). Local large deviation theorem for sums of i.i.d random vectors when the cramér condition holds in the whole space. *Probability and Mathematical Statistics 24*(2), 297–320.

[80] Kalagnanam, J. R. and U. M. Diwekar (1997). An efficient sampling technique for off-line quality control. *Technometrics 39*(3), 308–319.

[81] Kim, I. Y. and O. L. D. Weck (2005). Adaptive weighted-sum method for bi-objective optimization : Pareto front generation. *Structural and multidisciplinary optimization 29*(2), 149–158.

[82] Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi, et al. (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

[83] Kirsch, A. (2011). *An introduction to the mathematical theory of inverse problems*, Volume 120. Springer Science & Business Media.

[84] Larranaga, P. and J. A. Lozano (2002). *Estimation of distribution algorithms : A new tool for evolutionary computation*, Volume 2. Springer Science & Business Media.

[85] Lascaux, P. and R. Théodor (1987). *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, Volume 2. Masson Paris, France.

[86] Lawson, C. L. and R. J. Hanson (1974). *Solving least squares problems*, Volume 161. SIAM.

[87] Lehman, J. S., T. J. Santner, and W. I. Notz (2004). Designing computer experiments to determine robust control variables. *Statistica Sinica 14*(2), 571–590.

[88] Liaw, A. and M. Wiener (2002). Classification and regression by randomforest. *R news 2*(3), 18–22.

[89] Limbourg, P., E. D. Rocquigny, and G. Andrianov (2010). Accelerated uncertainty propagation in two-level probabilistic studies under monotony. *Reliability Engineering & System Safety 95*(9), 998–1010.

[90] Lourenço, H. R., O. C. Martin, and T. Stützle (2010). Iterated local search : Framework and applications. In *Handbook of Metaheuristics*, pp. 363–397. Springer.

[91] Lucquin, B., O. Pironneau, and M. Kern (1998). *Introduction to scientific computing*. Wiley Chichester.

[92] Luenberger, D. G. (1973). *Introduction to linear and nonlinear programming*, Volume 28. Addison-Wesley Reading, MA.

[93] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Volume 1, pp. 281–297. Oakland, CA, USA.

[94] Marler, R. T. and J. S. Arora (2010). The weighted sum method for multi-objective optimization : new insights. *Structural and multidisciplinary optimization 41*(6), 853–862.

[95] McKay, M. D. et al. (1995). *Evaluating prediction uncertainty*. US Nuclear Regulatory Commission.

[96] McKay, M. D., R. J. Beckman, and W. J. Conover (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*(2), 239–245.

[97] Miettinen, K. (2001). Some methods for nonlinear multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pp. 1–20. Springer.

[98] Miller, C. (2005). Search for level sets of functions using computer experiments. *Digital Repository@ Iowa State University, http ://lib. dr. iastate. edu*.

[99] Miller, R. G. J. (1997). *Beyond ANOVA : basics of applied statistics*. CRC Press.

[100] Morozov, V. A. (2012). *Methods for solving incorrectly posed problems*. Springer Science & Business Media.

[101] Moutoussamy, V. (2015). *Contributions to structural reliability : monotonicity constraints in numerical models*. ThÃ¨se de doctorat, UniversitÃ© Toulouse III.

[102] Myers, R. H. and D. C. Montgomery (2009). *Response surface methodology : process and product optimization using designed experiments*, Volume 705. John Wiley & Sons.

[103] Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability & Its Applications 9*(1), 141–142.

[104] Nakamura, G. and R. Potthast (2015). *Inverse Modeling*. 2053-2563. IOP Publishing.

[105] Nakayama, H. and Y. Sawaragi (1984). Satisficing trade-off method for multiobjective programming. In *Interactive decision analysis*, pp. 113–122. Springer.

[106] NguyenVan, T. (2006). *System engineering for collaborative data management systems : Application to design/simulation loops*. Ph. D. thesis, Ecole Centrale Paris.

[107] Osyczka, A. (1981). An approach to multicriterion optimization for structural design. Technical report, DTIC Document.

[108] Padulo, M. and M. D. Guenov (2011). Worst-case robust design optimization under distributional assumptions. *International Journal for Numerical Methods in Engineering 88*(8), 797–816.

[109] Plackett, R. L. and J. P. Burman (1946). The design of optimum multifactorial experiments. *Biometrika*, 305–325.

[110] Popelin, A.-L., R. Sueur, and N. Bousquet (2012). Encadrement et estimation de probabilités de défaillance dans un cadre monotone d'analyse de fiabilité structurale. *Congrès λμ 18, Tours, France*.

[111] Pukelsheim, F. (1993). *Optimal design of experiments*, Volume 50. siam.

[112] Rajabalinejad, M., L. E. Meester, P. V. Gelder, and J. K. Vrijling (2011). Dynamic bounds coupled with monte carlo simulations. *Reliability Engineering & System Safety 96*(2), 278–285.

[113] Rao, S. S. and S. S. Rao (2009). *Engineering optimization : theory and practice*. John Wiley & Sons.

[114] Resende, M. G. C. (2009). Greedy randomized adaptive search procedures. *Encyclopedia of optimization*, 1460–1469.

[115] Rocquigny, E. D. (2009). Structural reliability under monotony : Properties of form, simulation or response surface methods and a new class of monotonous reliability methods (mrm). *Structural Safety 31*(5), 363–374.

[116] Rubinstein, R. Y. and D. P. Kroese (2013). *The cross-entropy method : a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media.

[117] Ruppert, D. and M. P. Wand (1994). Multivariate locally weighted least squares regression. *The annals of statistics*, 1346–1370.

[118] Sabre, R. (2007). Plans d'expériences : Méthode de taguchi. *Techniques de l'ingénieur. Agroalimentaire 1*(F1006).

[119] Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications 145*(2), 280–297.

[120] Saltelli, A., M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola (2008). *Global sensitivity analysis : the primer*. John Wiley & Sons.

[121] Sergent, M., D. Dupuy, B. Corre, and M. Claeys-Bruno (2009). Comparaison de méthode criblage pour la simulation numérique. In *41èmes Journées de Statistique, SFdS, Bordeaux*.

[122] Shukla, P. K. (2007). On the normal boundary intersection method for generation of efficient front. In *Computational Science–ICCS 2007*, pp. 310–317. Springer.

[123] Siarry, P. and G. Dreyfus (1988). *La méthode du recuit simulé : théorie et applications*. IDSET.

[124] Simpson, T. W., D. K. Lin, and W. Chen (2001). Sampling strategies for computer experiments : design and analysis. *International Journal of Reliability and Applications 2*(3), 209–240.

[125] Simpson, T. W., J. D. Poplinski, P. N. Koch, and J. K. Allen (2001). Metamodels for computer-based engineering design : survey and recommendations. *Engineering with computers 17*(2), 129–150.

[126] Snecma (2012). *Méthodologie Conception Robuste*. Snecma. Manuel des Pratiques de Conception.

[127] Snecma (2014). *Appliquer le processus de calibration des jauges sur supports axisymétriques*. Snecma.

[128] Sobol', I. M. (1993). On sensitivity estimation for nonlinear mathematical models. *Mathematical Modelling ans Computational Experiments 1*, 407–414.

[129] Stein, M. L. (2012). *Interpolation of spatial data : some theory for kriging*. Springer Science & Business Media.

[130] Steuer, R. E. and E.-U. Choo (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming 26*(3), 326–344.

[131] Storlie, C. B. and J. C. Helton (2008). Multiple predictor smoothing methods for sensitivity analysis : Description of techniques. *Reliability Engineering & System Safety 93*(1), 28–54.

[132] Storlie, C. B., L. P. Swiler, J. C. Helton, and C. J. Sallaberry (2009). Implementation and evaluation of nonparametric regression procedures for sensitivity analysis of computationally demanding models. *Reliability Engineering & System Safety 94*(11), 1735–1763.

[133] Stutzle, T. and H. Hoos (2005). Stochastic local search : Foundations and applications.

[134] Süli, E. and D. F. Mayers (2003). *An introduction to numerical analysis*. Cambridge university press.

[135] Talbi, E.-G. (2009). *Metaheuristics : from design to implementation*, Volume 74. John Wiley & Sons.

[136] Tenenhaus, M. (1998). *La régression PLS : théorie et pratique*. Editions technip.

[137] Thévenin, J.-C. (2004). *Le turboréacteur, moteur des avions à réaction*. Association Aéronautique et Astronautique de France.

[138] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

[139] Tikhonov, A. N., A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola (2013). *Numerical methods for the solution of ill-posed problems*, Volume 328. Springer Science & Business Media.

[140] Trosset, M. W. (1996). Taguchi and Robust Optimization. Technical report, Department of Computational and Applied Mathematics, Rice University.

[141] Varga, R. S. (1962). Matrix iterative analysis. *Prentice Hall Series in Automatic Computations, Englewood Cliffs : Prentice-Hall, 1962 1*.

[142] Villa-Vialaneix, N., M. Follador, M. Ratto, and A. Leip (2012). A comparison of eight metamodeling techniques for the simulation of N_2o fluxes and N leaching from corn crops. *Environmental Modelling & Software 34*, 51–66.

[143] Wagner, T., M. Emmerich, A. Deutz, and W. Ponweiser (2010). On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature, PPSN XI*, pp. 718–727. Springer.

[144] Wang, G. G. and S. Shan (2007). Review of metamodeling techniques in support of engeneering design optimization. *Journal of Mechanical Design 129*(4), 370–380.

[145] Wang, G. G., L. Wang, and S. Shan (2005). Reliability assessment using discriminative sampling and metamodeling. Technical report, SAE Technical Paper.

[146] Wang, X., Y. Liu, and E. Antonsson (1999). Fitting functions to data in high dimensional design space. In *Proceedings of DETC*, Volume 99, pp. 1999.

[147] Watson, G. S. (1964). Smooth regression analysis. *Sankhyā : The Indian Journal of Statistics, Series A*, 359–372.

[148] Xie, Y.-L. and J. H. Kalivas (1997). Evaluation of principal component selection methods to form a global prediction model by principal component regression. *Analytica chimica acta 348*(1), 19–27.

[149] Zhang, W. H. and T. Gao (2006). A min–max method with adaptive weightings for uniformly spaced pareto optimum points. *Computers & structures 84*(28), 1760–1769.

Annexes

A Compléments pour la réduction de la dimension

Cette partie vient en complément de la Section 3 du Chapitre II.

A.1 Démonstration de l'Equation II.10

On considère que l'on dispose d'un plan à deux facteurs A et B, avec respectivement n_A et n_B niveaux. On veut montrer que l'Equation (II.10) du Chapitre II est vraie.

Démonstration. Comme le plan est complet, alors le nombre d'expériences du plan est $N = n_A n_B$ et la somme des carrés totale s'écrit

$$\begin{aligned} SC_T &= \sum_{k=1}^{N} (y_k - \bar{y})^2 \\ &= \sum_{k=1}^{N} y_k^2 - N\bar{y}^2 \\ &= \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} y_{A=i,B=j}^2 - N\bar{y}^2, \end{aligned}$$

avec $\bar{y} = \frac{1}{N} \sum_{i=1}^{n_A} \sum_{j=1}^{n_B} y_{A=i,B=j}$.

En utilisant (II.8) et (II.3), on établit que

$$\begin{aligned} SC_A &= \frac{N}{n_A}\sum_{i=1}^{n_A}(\bar{y}_{A=i}-\bar{y})^2 \\ &= \frac{N}{n_A}\sum_{i=1}^{n_A}\left(\frac{n_A}{N}\sum_{j=1}^{n_B}y_{A=i,B=j}-\bar{y}\right)^2 \\ &= \frac{N}{n_A}\sum_{i=1}^{n_A}\left[\frac{n_A^2}{N^2}\left(\sum_{j=1}^{n_B}y_{A=i,B=j}\right)^2-2\frac{n_A}{N}\bar{y}\sum_{j=1}^{n_B}y_{A=i,B=j}+\bar{y}^2\right] \\ &= \frac{n_A}{N}\sum_{i=1}^{n_A}\left(\sum_{j=1}^{n_B}y_{A=i,B=j}\right)^2-2\bar{y}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}y_{A=i,B=j}+N\bar{y}^2 \\ &= \frac{n_A}{N}\sum_{i=1}^{n_A}\left(\sum_{j=1}^{n_B}y_{A=i,B=j}\right)^2-N\bar{y}^2. \end{aligned} \tag{A.1}$$

De même, on a que

$$SC_B = \frac{n_B}{N}\sum_{j=1}^{n_B}\left(\sum_{i=1}^{n_A}y_{A=i,B=j}\right)^2-N\bar{y}^2. \tag{A.2}$$

En utilisant (II.9), (II.5) et la propriété d'orthogonalité du plan factoriel, qui implique que pour

un plan à deux facteurs, $\bar{y}_{A=i,B=j} = y_{A=i,B=j}$, on établit ensuite que

$$\begin{aligned}
SC_{AB} &= \frac{N}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(y_{A=i,B=j} - \frac{n_A}{N}\sum_{j=1}^{n_B} y_{A=i,B=j} - \frac{n_B}{N}\sum_{i=1}^{n_A} y_{A=i,B=j} + \bar{y}\right)^2 \\
&= \frac{N}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} y_{A=i,B=j}^2 + \frac{n_A}{N n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(\sum_{j=1}^{n_B} y_{A=i,B=j}\right)^2 + \frac{n_B}{N n_A}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(\sum_{i=1}^{n_A} y_{A=i,B=j}\right)^2 \\
&\quad + N\bar{y}^2 - \frac{2}{n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(y_{A=i,B=j}\sum_{j=1}^{n_B} y_{A=i,B=j}\right) - \frac{2}{n_A}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(y_{A=i,B=j}\sum_{i=1}^{n_A} y_{A=i,B=j}\right) \\
&\quad + \frac{2N}{n_A n_B}\bar{y}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} y_{A=i,B=j} + \frac{2}{N}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(\sum_{j=1}^{n_B} y_{A=i,B=j}\sum_{i=1}^{n_A} y_{A=i,B=j}\right) \\
&\quad - \frac{2}{n_B}\bar{y}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(\sum_{j=1}^{n_B} y_{A=i,B=j}\right) - \frac{2}{n_A}\bar{y}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B}\left(\sum_{i=1}^{n_A} y_{A=i,B=j}\right) \\
&= \frac{N}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} y_{A=i,B=j}^2 + \left(\frac{n_A}{N} - \frac{2}{n_B}\right)\sum_{i=1}^{n_A}\left(\sum_{j=1}^{n_B} y_{A=i,B=j}\right)^2 \\
&\quad + \left(\frac{n_B}{N} - \frac{2}{n_A}\right)\sum_{j=1}^{n_B}\left(\sum_{i=1}^{n_A} y_{A=i,B=j}\right)^2 + \frac{2N^2}{n_A n_B}\bar{y}^2 - N\bar{y}^2. \qquad \text{(A.3)}
\end{aligned}$$

La somme de (A.1), (A.2) et (A.3) donne

$$\begin{aligned}
SC_A + SC_B + SC_{AB} = \frac{N}{n_A n_B}\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} y_{A=i,B=j}^2 &+ 2\left(\frac{n_A}{N} - \frac{1}{n_B}\right)\sum_{i=1}^{n_A}\left(\sum_{j=1}^{n_B} y_{A=i,B=j}\right)^2 \\
&+ 2\left(\frac{n_B}{N} - \frac{1}{n_A}\right)\sum_{j=1}^{n_B}\left(\sum_{i=1}^{n_A} y_{A=i,B=j}\right)^2 - \left(\frac{2N^2}{n_A n_B} - 3N\right)\bar{y}^2.
\end{aligned}$$

Comme $N = n_A n_B$, alors $\frac{n_A}{N} - \frac{1}{n_B} = \frac{n_B}{N} - \frac{1}{n_A} = 0$ et donc

$$\begin{aligned}
SC_A + SC_B + SC_{AB} &= \sum_{i=1}^{n_A}\sum_{j=1}^{n_B} y_{A=i,B=j}^2 - 2N\bar{y}^2 \\
&= SCT
\end{aligned}$$

Ceci implique donc que $SC_R = 0$. On a donc bien démontré l'Equation (II.10). ■

A.2 Test de Student

Dans les études de criblage, des tests d'hypothèses sont menés dans le but de tester la significativité des effets des facteurs. Nous avons vu que le test de Fisher était utilisé pour les facteurs à 2 niveaux. Le test de Student va tester plus précisément la significativité d'un niveau pour les facteurs à plus de 2 niveaux. Ceci permet de déceler des non-linéarités dans la relation entre la sortie et les entrées.
Prenons par exemple un cas à quatre facteurs (A, B, C et D) à 3 niveaux chacun. Le graphe des effets est donné à la Figure A.1. La sortie semble avoir un comportement relativement linéaire

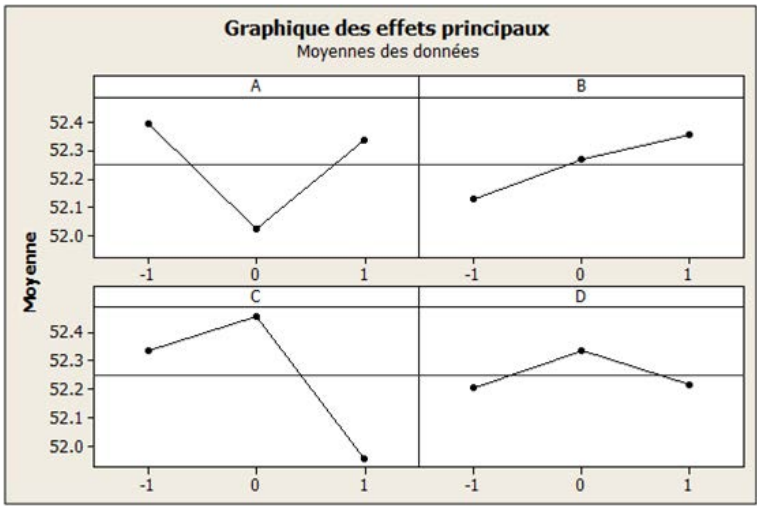

Figure A.1 – Graphe des effets pour quatre facteurs à 3 niveaux

par rapport aux facteurs B et D. Par contre, les facteurs A et B présentent clairement une non-linéarité.

L'hypothèse testée est H_0 : « l'effet de la modalité est nul ». La statistique de test pour la modalité i du facteur A est la suivante

$$t_A^i = \frac{e_A^i}{s_A^i / \sqrt{\frac{N}{n_A}}}, \tag{A.4}$$

où e_A^i est l'effet du facteur A pour le niveau i, n_A le nombre de niveaux de A, N le nombre d'expériences dans le plan et s_A^i l'écart-type estimé des valeurs de y lorsque A est au niveau i :

$$s_A^i = \sqrt{\frac{n_A}{N - n_A} \sum_{k=1^N} \left(y_{k|A=i} - \bar{y}_{A=i} \right)^2}$$

La statistique observée t_A^i est comparée au fractile de la loi de Student $t(\frac{N}{n_A} - 1, 1 - \frac{\alpha}{2})$ (cf Figure A.2. La zone de rejet de l'hypothèse H_0 est $\{|t_A^i| > t(\frac{N}{n_A} - 1, 1 - \frac{\alpha}{2})\}$. Plus $|t_A^i|$ est grand,

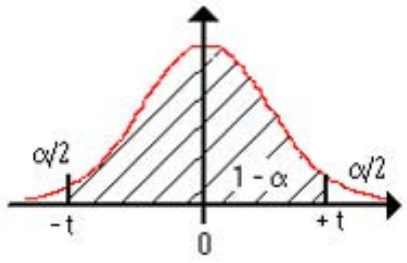

Figure A.2 – Loi de Student et son fractile d'ordre 0.95 ($\alpha = 5\%$)

plus l'effet de la modalité i du facteur A est significatif.

B Complément des méthodes de méta-modélisation

B.1 Réseaux de neurones

La méta-modélisation par réseaux de neurones est un cas particulier de régression linéaire généralisée où l'inverse de la fonction lien est une fonction sigmoïde

$$g(\hat{y}) = -T \log\left(\frac{1}{\hat{y}} - 1\right), \tag{B.5}$$

où T est le paramètre de pente de la sigmoïde, défini par l'utilisateur. Comme dans la régression linéaire généralisée, la sortie transformée a une relation linéaire avec les entrées :

$$g(\hat{y}) = \sum_{i=1}^{d} w_i x_i + \beta,$$

où les w_i et β sont les paramètres de la régression.
Une autre définition, proposée par (Simpson et al., 2001) consiste à considérer que les modèles de régression linéaire multiple sont en fait des neurones. Les entrées du système, $\{x_1, \ldots, x_d\}$, sont alors les entrées du neurone, les coefficients de régression w_i sont les poids du neurone et β est désigné comme le « biais » du neurone. Ainsi, (B.5) est équivalent à (B.6)

$$\hat{y} = \hat{f}(x) = \frac{1}{1 + e^{-\frac{\eta}{T}}}, \tag{B.6}$$

où $\eta = \sum_{i=1}^{d} w_i x_i + \beta$, avec β la valeur de biais du neurone. La fonction $\hat{f}$ est alors appelée fonction d'activation. Elle prend ici une forme sigmoïde mais on trouve des fonctions d'activation de deux autres types : la fonction tangente hyperbolique et la fonction de Heaviside.
Un réseau de neurones est un assemblage de neurones selon une architecture. Les réseaux les plus simples sont les perceptrons (neurone unique) (Figure B.3(a)) puisqu'ils sont mono-couches avec une seule sortie à laquelle toutes les entrées sont connectées. Les réseaux les plus utilisés sont les architectures prédictives multi-couches (MLP) (Figure B.3(b)). Dans une architecture

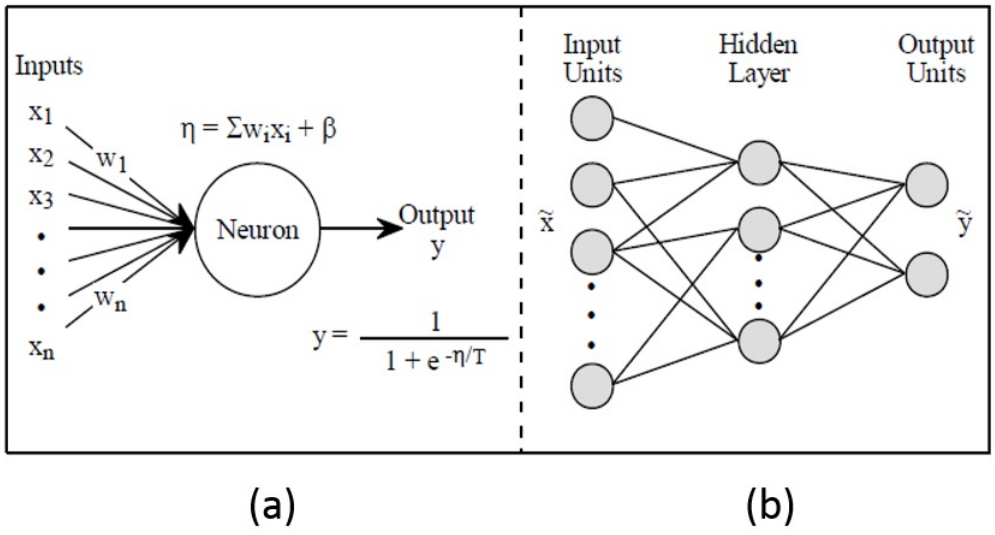

Figure B.3 – Schématisation d'un neurone (a) et d'une architecture (b)

multi-couches, les entrées de chaque couche sont les sorties de la couche précédente. Ainsi, les N_i neurones de la couche i prennent en entrée les N_{i-1} neurones de la couche précédente. Pour chaque couche, des poids sont attribués aux entrées, on parle de poids synaptiques. Chaque neurone de niveau i effectuera une somme pondérée des N_{i-1} neurones. C'est la fonction d'activation qui permet d'introduire une non-linéarité dans le fonctionnement du neurone.

La construction d'un réseau de neurones repose sur deux actions principales : spécifier l'architecture et former le réseau de neurones à partir d'un ensemble d'apprentissage. En statistiques, ceci est équivalent à spécifier un modèle de régression et estimer les paramètres du modèle sachant un ensemble de données. La formation du réseau de neurones consiste à mettre à jour tous les poids w_i dans l'architecture, à chaque itération. Ceci est habituellement fait par rétro-propagation, ce qui nécessite N points d'apprentissage $\{(x^1, y_1), \ldots, (x^N, y_N)\}$, où

$x^p = (x_1^p, \ldots, x_d^p)$. Pour un réseau dont la sortie est y, l'erreur totale du système est

$$E = \sum_p (y_p - \hat{y}_p)^2,$$

où $\hat{y}_p$ est l'estimation de la sortie, obtenue avec le réseau à partir de l'entrée x^p. Les poids sont alors ajustés proportionnellement à

$$\frac{\partial E}{\partial y} \frac{\partial y}{\partial w_{ij}}.$$

On parle de rétropropagation du gradient de l'erreur, qui consiste à corriger les erreurs selon l'importance des poids dans ces erreurs. Ainsi, les poids qui ont contribué à une erreur importante seront modifiés plus significativement que ceux qui ont contribué à une petite erreur.

Les réseaux de neurones sont bien adaptés pour l'approximation des fonctions déterministes dans les applications du type régression. Ainsi, toutes les fonctions continues bornées peuvent être représentées, avec une précision arbitraire, par un réseau à deux couches. Le théorème de Cybenko dit même que n'importe quelle fonction peut être approximée avec une précision arbitraire grâce à un réseau à 3 couches, à condition que le nombre de neurones dans les couches cachées soit suffisant. Ces méthodes ne permettent pas de prendre en compte l'aléatoire dans le processus, le but étant l'approximation de fonctions. Lorsque le nombre d'entrées devient important, la construction du réseau peut être coûteux en temps de calculs.
Ces méthodes ont notamment été utilisées dans (Villa-Vialaneix et al., 2012) et (Simpson et al., 2001). Dans (Aleksander and Morton, 1990), les auteurs exposent et illustrent les derniers développements en matière de réseaux de neurones.

B.2 Méthodes d'interpolation

B.2.1 Réseaux RBF

Les réseaux de type MLP (Multi-Layer Perceptron) calculent une combinaison linéaire des entrées, c'est-à-dire que le neurone renvoie le produit scalaire entre le vecteur des entrées et le vecteur des poids. D'autres stratégies peuvent être choisies comme les réseaux de type RBF (Radial Basis Function) qui calculent la distance entre les entrées. Dans ce cas, le neurone renvoie la norme euclidienne du vecteur issu de la différence vectorielle entre les vecteurs d'entrées.

Il s'agit de réseaux mono-couches pour lesquels

$$\hat{f}(x) = \sum_{i=1}^{d} w_i \phi(||x - x_i||),$$

où ϕ peut être de différentes formes :

- gaussienne : $\phi(r) = e^{(-\epsilon r)^2}$, ϵ est un facteur d'échelle,
- multi-quadratique : $\phi(r) = \sqrt{1 + (\epsilon r)^2}$,
- inverse quadratique : $\phi(r) = \frac{1}{1+(\epsilon r)^2}$,
- inverse multi-quadratique : $\phi(r) = \frac{1}{\sqrt{1+(\epsilon r)^2}}$,
- spline poly-harmonique : $\phi(r) = r^k(1_{\{k\ impair\}} + \ln r 1_{\{k\ pair\}}, k \in \mathbb{N}^*$,
- spline à plaque mince : $\phi(r) = r^2 \ln r$.

Ces méthodes sont construites dans un but d'interpolation. Il n'y a donc pas d'erreur d'apprentissage mais une grande quantité de points d'apprentissage peut poser problème. L'échantillon doit donc être choisi de façon à couvrir le domaine : l'idéal est de prendre des points équidistants.
Ces méthodes ont été utilisées dans (Jin et al., 2001) et (Wang and Shan, 2007).

B.2.2 Krigeage

Comme beaucoup de codes numériques sont déterministes et donc ne sont pas soumis à l'erreur de mesure, les mesures usuelles de l'incertitude issue des résidus des moindres carrés n'ont pas de sens évident. Ainsi, certains statisticiens ont suggéré de modéliser les réponses du système comme une combinaison d'un modèle polynomial plus des écarts

$$\hat{f}(x) = g(x) + Z(x)$$

où g est une fonction polynomiale connue et $Z(x)$ la réalisation d'un processus gaussien de moyenne nulle, de variance σ^2 et de covariance non nulle. Le terme $g(x)$ peut être une constante connue (krigeage simple), une constante inconnue (krigeage ordinaire) ou une combinaison linéaire de fonctions de base données (krigeage universel). Le terme $Z(x)$ crée des variations locales de telle sorte que le modèle de krigeage interpole les N points de l'échantillon d'apprentissage. La matrice de covariance de $Z(x)$ est donnée par

$$Cov[Z(x^i), Z(x^j)] = c(x^i, x^j) = \sigma^2 R(x^i, x^j),$$

où $R(x^i, x^j)$ est la fonction de corrélation entre deux points x^i et x^j de l'échantillon d'apprentissage. La fonction de corrélation doit être spécifiée par l'utilisateur et choisie symétrique et définie positive. Il existe plusieurs choix possibles pour la fonction de corrélation :

- exponentielle : $R(x^i, x^j) = \exp\left(-\sum_{k=1}^{d} \theta_k |x_k^i - x_k^j|\right)$,
- gaussienne : $R(x^i, x^j) = \exp\left(-\sum_{k=1}^{d} \theta_h^2 |x_k^i - x_k^j|^2\right)$,
- Matern : $R(x^i, x^j) = \left(\frac{1}{2^{\alpha-1}\Gamma(\alpha)}\right)^d \prod_{k=1}^{d} \left(\frac{|x_k^i - x_k^j|}{\theta_k}\right)^\alpha K_\alpha \left(\frac{|x_k^i - x_k^j|}{\theta_k}\right)$, avec $\alpha > 0$ et K_α la fonction de Bessel. On considère souvent le cas où $\alpha = p + 1/2$, $p \in \mathbb{N}$. Si $p = 0$, on retrouve le cas exponentiel. Les cas $p = 1$ et $p = 2$ sont respectivement appelés Matern3/2 et Matern5/2. Notons que la valeur de p est directement liée à la régularité du processus.

Il y a donc 2 types de paramètres à estimer, les paramètres de la moyenne notés β et le paramètre de variance σ^2 ainsi que des hyper-paramètres qui sont les paramètres θ de la covariance. L'estimation se fait par maximum de vraisemblance grâce aux N observations de l'échantillon d'apprentissage.

La construction du méta-modèle par processus gaussien se fait en 4 étapes :

1. choix de la fonction moyenne,
2. choix de la fonction de covariance,
3. estimation des paramètres,
4. construction du méta-modèle grâce au processus gaussien conditionnel aux points de l'échantillon d'apprentissage
    - le prédicteur est la moyenne du processus
    - l'erreur quadratique du prédicteur est la variance du processus.

Un méta-modèle de krigeage a une erreur d'apprentissage nulle puisqu'il est construit comme un interpolateur exact des points de l'échantillon d'apprentissage. L'estimateur obtenu est le meilleur estimateur linéaire sans biais (BLUE). En outre, l'estimation des paramètres par maximum de vraisemblance rend la méthode facile à utiliser en grande dimension (beaucoup de variables d'entrée, <50). La formulation analytique n'est pas simple mais est très flexible et permet d'évaluer rapidement de nouveaux points. Ensuite, la méthode est bien adaptée aux applications déterministes et n'est pas limitée par les hypothèses sur la nature de l'erreur aléatoire dans les observations. Enfin, il est possible d'utiliser cette méthode dans une planification adaptative où le modèle est amélioré à chaque point ajouté à l'échantillon d'apprentissage.
Les principaux inconvénients portent sur le choix des fonctions moyenne et de covariance et sur l'estimation des paramètres de régression et de covariance en grande dimension. En outre, la méthode n'est pas adaptée lorsqu'on a beaucoup de données. En effet, l'inversion de la matrice

de covariance devient coûteuse et l'optimisation de la vraisemblance difficile. Enfin, le prédicteur obtenu a tendance à lisser les données.
Ces estimateurs ont été utilisés dans (Barton, 1994), (Simpson et al., 2001), (Storlie et al., 2009), (Wang and Shan, 2007), (Villa-Vialaneix et al., 2012) et (Jin et al., 2001). Un livre a également été proposée (Stein, 2012).

B.3 Méthodes de régression non-paramétrique

Nous présentons ici une méthode univariée basée sur les splines de lissage et neuf méthodes multivariées très différentes.

B.3.1 Projection sur des bases usuelles

Les transformations utilisées sur les entrées x forment un ensemble $\mathcal{F}$, appelé dictionnaire, de fonctions non-linéaires

$$\hat{f}(x) = \sum_{g \in \mathcal{F}} \beta_g g(x).$$

Il existe trois approches de construction du dictionnaire de fonctions :

- par restriction : choix basé sur des connaissances du modèle (par la construction du modèle, à partir d'avis d'expert),
- par sélection : choix d'un large dictionnaire dans lequel seules les fonctions les plus significatives sont retenues,
- par régularisation : choix d'un large dictionnaire et mise en place de contraintes sur les coefficients du modèle.

La taille M du dictionnaire est un paramètre de lissage permettant d'assurer l'équilibre biais-variance.

Parmi les bases usuelles, on trouve

- les polynômes (cf Chapitre II),
- la base trigonométrique (base orthonormée de $\mathbb{L}^2[0,1]$),
- les bases d'ondelettes (utiles pour les fonctions à grandes irrégularités locales comme en traitement du signal),
- les splines.

Voyons plus en détail le cas des splines dans la section qui suit.

B.3.2 Méthode univariée des splines de lissage

Parmi les méthodes de lissage de nuages de points, on trouve les splines cubiques de lissage. Dans le cas unidimensionnel ($d = 1$), la fonction $\hat{f}$ minimise

$$\sum_{i=1}^{N}[y_i - \hat{f}(x^i)]^2 + \lambda \int_a^b \left[\frac{d^2\hat{f}(x)}{dx^2}\right]^2 dx$$

sur toutes les fonctions $\hat{f}$ différentiables, où $a \leq \min\{x^i : 1 \leq i \leq N\}$, $\max\{x^i : 1 \leq i \leq N\} \leq b$ et λ est une constante. Il y a une unique solution explicite à ce problème de minimisation. Cette fonction est une spline cubique avec des nœuds (endroits où la structure de la spline change) en les valeurs observées pour x. Une spline cubique est un polynôme cubique sur tout intervalle défini par les nœuds adjacents. Elle a deux dérivées continues, sa troisième dérivée est une fonction étagée avec des sauts au niveau des nœuds. La constante λ joue ici le rôle de paramètre de lissage dont la valeur peut être choisie par validation croisée. Dans certaines cas, les splines cubiques de lissage peuvent présenter des comportements similaires à la méthode LOESS.
Cette méthode est proposée dans (Barton, 1994) et (Storlie and Helton, 2008). Elle est également étudiée dans (Gu, 2013).

B.3.3 Modèles additifs

En multidimensionnel, on trouve également les modèles additifs (GAM - Generalized Additive Model), pour lesquels la fonction f est définie comme

$$\hat{f}(x) = \sum_{j=1}^{d} f_j(x_j),$$

où les f_j sont des fonctions de lissage déterminées dans le processus. Les effets des variables indépendantes sont donc additifs, ce qui ressemble à une régression linéaire multiple d'ordre 1. La différence est que f n'est pas supposée linéaire par rapport aux variables d'entrée. Cette représentation n'est cependant pas complètement générale puisqu'elle ne fait pas intervenir les interactions éventuelles entre les entrées. La procédure de construction des modèles additifs, appelée algorithme de backfitting, est décrite dans (Storlie et al., 2009).
Ces modèles sont très pratiques pour estimer des fonctions avec un comportement non linéaire complexe. Ils fonctionnent également bien en grande dimension.

Les modèles additifs peuvent être vus comme les modèles les plus simples parmi les splines de lissage ANOVA. En effet, la décomposition ANOVA de f en termes de dimension croissante s'écrit

$$f(x) = f(x_1, \ldots, x_d) = f_0 + \sum_j f^{(j)} + \sum_{k>j} f^{(jk)} + \ldots + f^{(12\ldots k)},$$

où chaque $f^{(j)}$ est fonction uniquement de x_j. Les termes $f^{(j)}$ représentent le modèle additif simple tandis que les termes d'ordres supérieurs $f^{(jk)}, \ldots, f^{(12\ldots k)}$ désignent les interactions.
Ces modèles ont été utilisés dans (Hastie and Tibshirani, 1990), (Storlie et al., 2009) et (Storlie and Helton, 2008).

B.3.4 Méthodes COSSO et ACOSSO

Supposant que $f \in \mathcal{H}$, avec $\mathcal{H}$ un espace de Hilbert particulier, la méthode COSSO (Component Selection and Shrinkage Operator) permet d'estimer f en minimisant chaque terme de l'ANOVA avec une pénalisation LASSO de la somme des normes. L'estimateur de f minimise donc

$$\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x^i)\right)^2 + \lambda \sum_{j=1}^{q} ||P^j \hat{f}||_{\mathcal{H}},$$

où $P^j\hat{f}$ est la projection orthogonale de $\hat{f}$ sur H_j tel que $H = \{1\} \oplus \{\oplus_{j=1}^{Q} H_j\}$, λ est un paramètre de lissage et q contient tous les termes de l'ANOVA susceptibles d'être inclus dans $\hat{f}$. La méthode ACOSSO est une amélioration de la méthode COSSO où $\hat{f}$ minimise

$$\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x^i)\right)^2 + \lambda \sum_{j=1}^{q} ||w_j P^j \hat{f}||_{\mathcal{H}},$$

où les w_j sont des poids qui dépendent d'une estimation initiale de f obtenue avec COSSO par exemple.
Ces méthodes sont rapides seulement si le nombre d'observations est très petit. De plus, l'hyper-paramètre λ est réglé deux fois : une première fois pour obtenir les poids w_j, une seconde fois pour obtenir l'estimateur final. Enfin, le temps nécessaire pour obtenir de nouvelles prédictions peut être grand, selon la taille des échantillons d'apprentissage et de test.
Ces méthodes sont proposées dans (Villa-Vialaneix et al., 2012) et (Storlie et al., 2009).

B.3.5 Projection par directions révélatrices

Les méthodes de projection par directions révélatrices (PPR - Projection Pursuit Regression) font également partie des lisseurs multivariés. Ces méthodes font intervenir la réduction de

dimension et la modélisation additive en utilisant une fonction $\hat{f}$ de la forme

$$\hat{f}(x) = \sum_{s=1}^{S} g_s(\alpha_s x),$$

où $\alpha_s = [\alpha_{1s}, \ldots, \alpha_{ks}]$, avec α_s et α_t orthogonaux pour $s \neq t$, $x = [x_1, \ldots, x_d]^t$, $\alpha_s x$ correspond à la combinaison linéaire des éléments de x et g_s est une fonction arbitraire unidimensionnelle des combinaisons linéaires des prédicteurs.
L'expression de $\hat{f}$ est un modèle additif où les quantités $\alpha_s x$ remplacent les éléments x_j de x en tant que prédicteurs. De plus, cette expression implique une réduction de la dimension puisque S est habituellement plus petit que d. Contrairement au cas des modèles additifs, cette estimation de f permet les interactions entre les variables. Les α_s et g_s sont estimés par la minimisation de sommes successives décrites dans (Storlie and Helton, 2008).
La projection par directions révélatrices peut représenter des situations très générales impliquant de la non-linéarité et des interactions. De plus, la méthode évite le fléau de la dimension en utilisant des termes de projection et une modélisation additive. Par contre, la méthode a tendance à sur-ajuster les données en incluant de fausses variables dans le modèle.
Ces méthodes sont ont proposées dans (Friedman and Stuetzle, 1981) puis utilisées entre autres dans (Storlie et al., 2009), (Friedman, 1991) et (Storlie and Helton, 2008).

B.3.6 Régression par partitionnement récursif

La régression par partitionnement récursif fait appel à des arbres de régression qui séparent les données en sous-groupes homogènes. La construction d'un arbre de régression a pour but de trouver une série de divisions. Chaque division est déduite d'une des d variables d'entrée, x_j, et d'un seuil τ pour séparer l'échantillon d'apprentissage en deux. Les deux sous-échantillons sont appelés des nœuds : $\{i : x_j^i < \tau\}$ et $\{i : x_j^i \geq \tau\}$. La division d'un nœud $\mathcal{N}$ est choisi parmi toutes les divisions possibles par une minimisation. Il s'agit de minimiser la somme de l'homogénéité des deux nœuds enfants $\mathcal{N}_c^1$ et $\mathcal{N}_c^2$

$$\sum_{i \in \mathcal{N}_c^i} \left(y_i - \bar{y}^{\mathcal{N}_c^1}\right)^2,$$

où $\bar{y}^{\mathcal{N}_c^1} = \frac{1}{|\mathcal{N}_c^1|} \sum_{\mathcal{N}_c^i} y_i$ est la moyenne des observations de la sortie appartenant à $\mathcal{N}_c^i$ (variance intra-nœud). La fonction f est alors estimée comme étant la moyenne de l'échantillon sur chaque

sous-groupe

$$\hat{f}(x) = \sum_{s=1}^{P} c_s I_s(x),$$

où $c_s = \sum_{x^i \in \mathcal{A}_s} \frac{y_i}{card(\mathcal{A}_s)}$, avec $\mathcal{A}_s, s = 1, \ldots, P$ sont les ensembles disjoints sur lesquels les valeurs observées $x^i, i = 1, \ldots, S$ ont été partitionnées et I_s est la fonction indicatrice valant 1 si l'élément est dans $\mathcal{A}_s$. L'estimateur de f est donc une fonction constante par morceaux, appelée aussi fonction simple.
Les arbres de régression peuvent être généralisés en remplaçant c_s par une fonction linéaire. L'estimateur de f est alors défini par

$$\hat{f}(x) = \sum_{s=1}^{P} (\hat{\alpha}_s + \hat{\beta}_s x) I_s(x),$$

où $\hat{\alpha}_s + \hat{\beta}_s x$ est l'ajustement linéaire par moindres carrés des données associées à $\mathcal{A}_s$. Cet ajustement linéaire permet de réduire le nombre de sous-groupes par rapport à la méthodes des arbres de régression. La détermination des ensembles $\mathcal{A}_s$ dans ce cas est décrit dans (Storlie and Helton, 2008). La méthode est également plus performante quand la relation entre la sortie et les entrées est proche de la linéarité pour chaque sous-groupe $\mathcal{A}_s$. Par contre, l'interprétation de l'estimateur de f peut s'avérer moins évidente que dans le cas des sommes.
Cette méthode permet d'effectuer une sélection de variables. En effet, f peut dépendre globalement d'un grand nombre de variables mais sur chaque sous-groupe, elle ne dépend vraiment que de certaines d'entre elles. Cette capacité vient de la nature récursive du partitionnement et fournit donc des sélections locales de variables dont découle naturellement la sélection globale. Cette propriété n'est en général pas satisfaite pour le partitionnement récursif basé sur les fonctions linéaires. Cet estimateur est utilisé dans (Storlie et al., 2009), (Friedman and Stuetzle, 1981) et (Storlie and Helton, 2008).

B.3.7 Forêts aléatoires

Les forêts aléatoires (RF - Random Forest) sont des méthodes non-paramétriques qui utilisent également les arbres de régression. La régression par partitionnement récursif se fait avec un seul arbre de régression, ce qui rend la méthode très sensible à des petits changements de l'échantillon d'apprentissage. Les forêts aléatoires font appel à un grand nombre d'arbres, T. Les modèles sont alors construits sur des échantillons aléatoires obtenus soit par bootstrap (échantillonnage aléatoire avec remise) soit par sous-échantillonnage (échantillonnage sans remise) sur les données d'apprentissage. L'algorithme consiste en deux étapes :

- on choisit aléatoirement m observations dans l'échantillon d'apprentissage : ce sous-ensemble est appelé l'échantillon « in-bag », le reste des observations constituent l'échantillon « out-of-bag » et sont utilisées pour vérifier l'erreur de l'arbre,
- pour chaque nœud de l'arbre, on sélectionne aléatoirement q variables parmi toutes les variables d'entrée possibles. Le meilleur partage est alors calculé sur la base de ces q variables pour les m observations choisies.

Tous les arbres de la forêt sont complètement construits une fois que les feuilles finales ne contiennent plus qu'une seule sortie observée. Une fois les T arbres de régression obtenus, la prédiction pour de nouvelles valeurs d'entrée x est égale à la moyenne des prédictions individuelles de x sur chaque arbre de la forêt.

Ces méthodes sont rapides car pratiquement insensibles à la dimension de l'échantillon d'apprentissage grâce à la sélection aléatoire des observations et des variables. La majorité du temps nécessaire est dû au nombre d'arbres requis pour stabiliser l'algorithme qui peut être grand. Le temps pour obtenir de nouvelles prédictions est également faible, cela dépend du nombre de prédictions à faire et du nombre d'arbres dans la forêt.
On trouve ces méthodes dans (Villa-Vialaneix et al., 2012), (Liaw and Wiener, 2002) et (Storlie et al., 2009).

B.3.8 Régression par amélioration de gradient

Dans les méthodes de régression par amélioration de gradient (GBR - Gradient Boosting Regression), une suite d'arbres simples est générée, où chaque arbre successif est construit pour prédire les résidus de l'arbre précédent. Contrairement aux forêts aléatoires où chaque arbre évolue jusqu'à ce que chaque groupe ne soit constitué que de quelques observations, cette méthode considère des arbres simples et un nombre restreint de groupes.
L'algorithme consiste en trois étapes :

- Ajuster un arbre de régression avec J nœuds sur l'ensemble des données d'origine $\{(x^i, y_i)\}_{i=1}^N$. Il s'agit de chercher les données pour la meilleure variable et le seuil le long de cette variable pour séparer les données. On répète ce processus sur chacun des deux sous-ensembles de données obtenus (un nœud) pour trouver la meilleure variable et le seuil pour faire une séparation sur seulement une de ces régions. On continue jusqu'à ce qu'on ait J nœuds. On appelle cet estimateur $\hat{f}_1$.
- Pour $j = 2, \ldots, N_t$, N_t étant le nombre d'arbres dans le développement
  - Ajuster un arbre de régression avec J nœuds à l'ensemble des données $\{(x^i, e_{j-1,i})\}_{i=1}^N$,

où $e_{j-1,i} = y_i - \sum_{l=1}^{j-1} \hat{f}_l(x^i)$ sont les résidus du modèle de l'itération précédente.

- On appelle cet estimateur $\hat{f}_j$.

- L'estimateur final est donné par

$$\hat{f}(x) = \sum_{j=1}^{N_t} \hat{f}_j(x).$$

La performance de l'algorithme peut être améliorée en ajoutant une régularisation ou un terme de pénalité au développement additif. Dans l'étape 2 de l'algorithme, les résidus seraient calculés comme $e_i = y_i - \sum_{l=1}^{j-1} \nu \hat{f}_l(x^i)$ et

$$\hat{f}(x) = \nu \sum_{j=1}^{N_t} \hat{f}_j(x)$$

serait utilisé comme estimateur final dans l'étape 3. Comme les forêts aléatoires, la méthode GBR travaille bien en grande dimension mais ne permet pas de faire de la sélection de variables. Ces méthodes ont été proposées dans (Friedman, 2001) et sont étudiées dans (Storlie et al., 2009) par exemple.

B.3.9 Méthode MARS

La procédure MARS (Multivariate Adaptive Regression Splines), proposée dans (Friedman, 1991) est une combinaison de la régression par splines, de l'ajustement de modèles pas à pas et de partitionnement récursif. Il s'agit de sélectionner de manière adaptative un ensemble de fonctions d'une base pour estimer f par une approche itérative progressive/régressive. Le modèle s'écrit

$$\hat{f}(x) = \sum_{m=1}^{M} a_m B_m(x),$$

où les a_m sont les coefficients obtenus en minimisant un critère de variance croisée généralisé (GCV) (moyenne quadratique des résidus de l'ajustement des données fois une pénalité prenant en compte la variance dépendant du nombre de fonctions M) et B_m les fonctions de la base représentées comme le produit de splines univariées d'ordre 1 ou cubiques. Pour plus de précisions, le lecteur intéressé pourra se référer à (Jin et al., 2002) et (Storlie and Helton, 2008). La méthode se fait de manière progressive ou de manière régressive. Dans la première, on part d'un modèle avec seulement un terme constant correspondant à la moyenne des valeurs de la sortie. La procédure consiste ensuite à ajouter successivement les fonctions de base au modèle. A chaque étape, la fonction choisie est celle qui maximise la réduction de l'erreur résiduelle. La procédure s'arrête lorsque l'évolution de l'erreur résiduelle est minimale ou lorsque le maximum

de termes est atteint. La recherche à chaque étape peut être relativement rapide en utilisant une technique de mise à jour des moindres carrés.
Dans la méthode régressive, l'idée est la même sauf que l'on part d'un modèle sur-ajusté et les termes sont retirés un à un selon le critère GCV.

Comme un effet de bord de la construction de ce modèle, il peut arriver que certaines fonctions de la base, correspondant à certaines variables d'entrée, n'apparaissent pas dans le modèle final. La méthode fait donc une sélection de variable automatique.
Dans (Jin et al., 2001), les auteurs affirment que les modèles obtenus sont plus flexibles que la régression linaire. Ils sont également simples à comprendre et à interpréter. En plus, la méthode peut considérer aussi bien des données continues que catégorielles. Elle tend à être meilleure que le partitionnement récursif pour les données numériques et est adaptée à de grands échantillons d'apprentissage. Les modèles MARS ne donnent pas des ajustements aussi bons que certaines autres méthodes utilisant les arbres de régression mais peuvent être construits beaucoup plus rapidement. Les prédictions sont également rapides à évaluer.

B.3.10 Machines à support de vecteur

La méthode SVM (Support Vector Machine), issue de la classification, est utilisée pour résoudre des problèmes de régression dans le cas de la prédiction de plusieurs variables de sorties réelles dépendantes. L'estimateur de f est choisi parmi la famille des fonctions

$$\hat{f} : x \in \mathbb{R}^k \mapsto \langle w | \phi(x) \rangle_{\mathcal{H}} + b,$$

où ϕ est une fonction de $\mathbb{R}^k$ dans un espace de Hilbert, $w \in \mathcal{H}$ et $b \in \mathbb{R}$ sont les paramètres à estimer à partir de l'échantillon d'apprentissage. L'approche originale pour déterminer w et b consiste à utiliser une fonction de perte comme critère de qualité pour la régression

$$L_\epsilon(x, y, \hat{f}) = \sum_{i=1}^{N} \max(|\hat{f}(x^i) - y_i| - \epsilon, 0).$$

Cette fonction de perte a la propriété d'éviter de considérer l'erreur de modélisation en un point quand elle est plus petite que ϵ. Son intérêt principal réside, comparée à l'erreur quadratique standard, dans sa robustesse. La régression SVM est basée sur la minimisation de cette fonction

de perte sur l'échantillon d'apprentissage en pénalisant la complexité de $\hat{f}$

$$\arg\min_{w,b} L_\epsilon(x, y, \hat{f}) + \frac{1}{C}||w||^2_{\mathcal{H}},$$

où $||w||^2_{\mathcal{H}}$ est le terme de régularisation qui contrôle la complexité de $\hat{f}$ et C est le paramètre de régularisation. Quand C est petit, $\hat{f}$ est autorisé à prendre de grandes erreurs en faveur d'une petite complexité. Quand C est grand, $\hat{f}$ ne fait presque pas d'erreur d'apprentissage mais peut avoir une grande complexité et donc ne pas fournir de bonnes estimations à de nouvelles observations. Un bon choix est donc un compromis entre la précision et la complexité du méta-modèle. Davantage d'informations sur la méthode sont données dans (Villa-Vialaneix et al., 2012) et (Gunn et al., 1998).
Cette méthode est rapide s'il n'y a pas trop d'observations mais le temps de prédiction de nouveaux points peut être grand. En outre, des hyper-paramètres comme w et b doivent être réglés. Si l'échantillon d'apprentissage est trop grand, la validation croisée n'est pas adaptée. Il faut alors valider le modèle par validation simple, ce qui peut être moins précis et plus coûteux.

C Démonstration de la convergence de la méthode MRM

Quelle que soit la méthode utilisée, Monte-Carlo adaptatif, dichotomique, rectangles, il y a toujours convergence vers la solution S. En effet, à chaque itération, le domaine inexploré diminue, plus ou moins rapidement, jusqu'à être réduit à S. Le but de cette section est donc de démontrer que la suite $(\mathbb{U}_k)_{k\in\mathbb{N}}$ tend vers S quand k tend vers l'infini.

Preuve. On munit notre espace de départ de la tribu engendrée par ses ouverts et de la mesure de Lebesgue, ce qui nous donne l'espace mesuré $([-1,1]^d, \mathcal{B}_{[-1,1]^2}, \lambda)$. L'espace est également muni d'une distance D.
Comme $\lambda([-1,1]^d) = 2^d < \infty$, la mesure λ est finie et donc σ-finie sur $[-1,1]^d$. Il existe donc un recouvrement dénombrable de $[-1,1]^d$ par des sous-ensembles de mesure finie, c'est-à-dire qu'il existe une suite $(E_k)_{k\in\mathbb{N}}$ d'éléments de $\mathcal{B}_{[-1,1]^d}$, tous de mesure finie, tels que $[-1,1]^d = \bigcup_{k\in\mathbb{N}} E_k$.
Il en est de même pour $\mathbb{U}^+$ et $\mathbb{U}^-$ puisque $\mathbb{U} = \mathbb{U}^+ \cup \mathbb{U}^- \cup S$ avec $\lambda[S] = 0$ et $\lambda([-1,1]^d) = \lambda[\mathbb{U}^+] + \lambda[\mathbb{U}^-]$.
Nous allons démontrer que la suite $(\mathbb{U}^+_k)$ est un recouvrement de $\mathbb{U}^+$, c'est-à-dire que tout élément $x \in \mathbb{U}^+$ se trouve dans au moins un des éléments de $(\mathbb{U}^+_k)$.
Pour cela, nous procédons par l'absurde et considérons un point $x^* \in \mathbb{U}^+$ tel que $\forall k \in \mathbb{N}$, $x^* \notin \mathbb{U}^+_k$. La première hypothèse implique que $f(x^*) > 0$, c'est-à-dire : $\exists \epsilon > 0$ tel que $D(x^*, S) = \epsilon$. La

seconde hypothèse implique que, pour ce point x^*, on ne s'est jamais trouvé dans un ensemble de mesure non nulle au cours de l'algorithme : non seulement x^* ne s'est pas trouvé dans un tel espace ni aucun point qui lui serait inférieur au sens de l'ordre partiel.
On considère l'ensemble des points inférieurs à x^* : $H_{x^*} = \{x : x \preceq x^*\}$. On définit la boule de centre x^* et de rayon ϵ : $B(x^*, \epsilon) = \{x, D(x, x^*) < \epsilon\}$. La Figure C.4 illustre la démonstration dans le cas bi-dimensionnel. La zone jaune représente l'intersection entre la boule $B(x^*, \epsilon)$ et l'ensemble H_{x^*} contenant tous les points inférieurs à x^* au sens de l'ordre partiel. On note

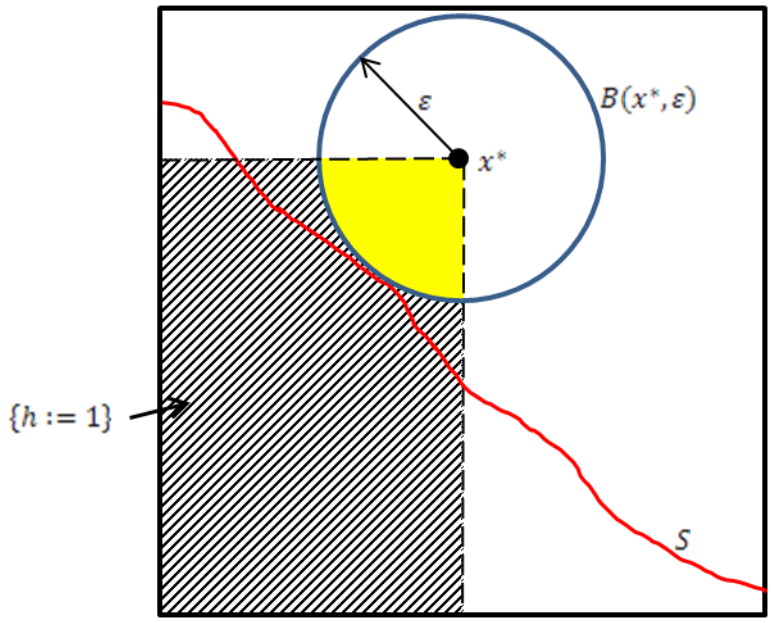

Figure C.4 – Illustration de la convergence de l'algorithme dans le cas bidimensionnel

$E_{x^*} = H_{x^*} \cap B(x^*, \epsilon)$ l'intersection représentée à la Figure C.4. La mesure de cet ensemble représente le quart du volume de l'hypersphère formée par la boule $B(x^*, \epsilon)$. Donc,

$$\lambda(E_{x^*}) = \frac{\pi^{\frac{d+1}{2}} \epsilon^{d+1}}{4\Gamma(\frac{d+3}{2})},$$

où d est la dimension du problème et Γ est la fonction gamma telle que

$$\Gamma\left(\frac{d+3}{2}\right) = \Gamma\left(\frac{d+1}{2}+1\right) = \begin{cases} (\frac{k+1}{2})! \text{ si } k \text{ est impair} \\ \frac{1\times3\times\ldots\times(k+1)}{2^{\frac{d}{2}+1}} \text{ si } k \text{ est pair} \end{cases}$$

Comme $d > 0$, alors $\Gamma(\frac{d+3}{2}) > 0$. De plus, $\epsilon > 0$, donc $\lambda(E_{x^*}) > 0$. L'ensemble E_{x^*} est ainsi de mesure non nulle, il y a donc eu au moins un point tiré dans cet ensemble au cours

de l'algorithme. On remarque que x^* lui-même appartient à cet ensemble. Il y a donc une contradiction avec l'affirmation que pour ce point x^*, « on ne s'est jamais trouvé dans un ensemble de mesure non nulle au cours de l'algorithme », donc, pour $x^* \in \mathbb{U}^+$, $\exists k \in \mathbb{N}, x^* \in \mathbb{U}_k^+$. La même démonstration peut être effectuée pour $x^* \in \mathbb{U}^-$. Dans ce cas, seul l'ensemble H_{x^*} sera différent : $H_{x^*} = \{x : x \succeq x^*\}$, le reste de la démonstration sera inchangé.
Ceci est vrai quel que soit x^*, donc on a bien des recouvrements de $\mathbb{U}^+$ et $\mathbb{U}^+$:

$$\mathbb{U}^+ = \bigcup_{k\in\mathbb{N}} \mathbb{U}_k^+$$
$$\mathbb{U}^- = \bigcup_{k\in\mathbb{N}} \mathbb{U}_k^-$$

Il reste à démontrer que $lim_{k\to+\infty}\mathbb{U}_k = S$. On sait que $\forall k \in \mathbb{N}, \mathbb{U}_k^+ \subset \mathbb{U}_{k+1}^+$ et $\mathbb{U}_k^- \subset \mathbb{U}_{k+1}^-$. Ceci vient du fait qu'à l'étape $k+1$ durant laquelle un point x a été simulé dans $\mathbb{U}_k$, on rajoute au domaine rejeté $\mathbb{U}_k^+$ (resp. $\mathbb{U}_k^-$) si $f(x) > 0$ (resp. $f(x) < 0$), l'ensemble des points supérieurs à x (resp. inférieurs à x) selon l'ordre partiel : $H_x = \{y : y \succeq x\}$ (resp. $H_x = \{y : y \preceq x\}$). Ainsi, $\mathbb{U}_{k+1}^+ = \mathbb{U}_k^+ \cup H_x$ (resp. $\mathbb{U}_{k+1}^- = \mathbb{U}_k^- \cup H_x$). Les suites $(\mathbb{U}_k^+)_{k\in\mathbb{N}}$ et $(\mathbb{U}_k^-)_{k\in\mathbb{N}}$ sont croissantes. On a donc

$$\lim_{k\to+\infty} \mathbb{U}_k^+ = \bigcup_{k\in\mathbb{N}} \mathbb{U}_k^+ = \mathbb{U}^+$$
$$\lim_{k\to+\infty} \mathbb{U}_k^- = \bigcup_{k\in\mathbb{N}} \mathbb{U}_k^- = \mathbb{U}^-$$

Comme on a : $\forall k \in \mathbb{N}, \mathbb{U}_k = \mathbb{U} \setminus (\mathbb{U}_k^+ \cup \mathbb{U}_k^-)$, alors la suite $(\mathbb{U}_k)_{k\in\mathbb{N}}$ est décroissante. Ainsi, $\lim_{k\to+\infty} \mathbb{U}_k = \bigcap_{k\in\mathbb{N}} \mathbb{U}_k$. Or

$$\begin{aligned}
\bigcap_{k\in\mathbb{N}} \mathbb{U}_k &= \bigcap_{k\in\mathbb{N}} \mathbb{U} \setminus (\mathbb{U}_k^+ \cup \mathbb{U}_k^-) \\
&= \bigcap_{k\in\mathbb{N}} (\mathbb{U} \setminus \mathbb{U}_k^+) \cap (\mathbb{U} \setminus \mathbb{U}_k^-) \\
&= \mathbb{U} \setminus [(\bigcup_{k\in\mathbb{N}} \mathbb{U}_k^+ = \mathbb{U}^+) \cup (\bigcup_{k\in\mathbb{N}} \mathbb{U}_k^+ = \mathbb{U}^-)] \\
&= \mathbb{U} \setminus (\mathbb{U}^+ \cup \mathbb{U}^-) \\
&= S
\end{aligned}$$

Donc $lim_{k\to+\infty}\mathbb{U}_k = S$, la méthode converge bien vers la solution S. ■

L'objectif de ce travail est de proposer une nouvelle démarche pour améliorer et accélérer les études de dimensionnement des pièces de turboréacteurs en avant-projets. Il s'agit de fournir une méthodologie complète pour la conception robuste sous contraintes. Cette méthodologie consiste en trois étapes : la réduction de la dimension et la méta-modélisation, la conception robuste sous contraintes puis la résolution de problèmes inverses sous contraintes. Ce sont les trois principaux sujets abordés dans cette thèse.

La réduction de la dimension est un pré-traitement indispensable à toute étude. Son but est de ne conserver, pour une sortie choisie du système, que les entrées influentes. Ceci permet de réduire la taille du domaine d'étude afin de faciliter la compréhension du système et diminuer les temps de calculs des études. Les méthodes de méta-modélisations contribuent également à ces deux objectifs. L'idée est de remplacer le code de calculs coûteux par un modèle rapide à évaluer et qui représente bien la relation entre la sortie étudiée et les entrées du système.

La conception robuste sous contraintes est une optimisation bi-objectifs où les différentes sources d'incertitudes du système sont prises en compte. Il s'agit, dans un premier temps, de recenser et modéliser les incertitudes puis de choisir une méthode de propagation de ces incertitudes dans le code de calculs. Ceci permet d'estimer les moments (moyenne et écart-type) de la loi de la sortie d'intérêt. L'optimisation de ces moments constitue les deux objectifs de la conception robuste. En dernier lieu, il s'agit de choisir la méthode d'optimisation multi-objectifs qui sera utilisée pour obtenir l'optimum robuste sous contraintes.

La partie innovante de cette thèse porte sur le développement de méthodes pour la résolution de problèmes inverses mal posés. Ce sont des problèmes pour lesquels il peut y avoir une infinité de solutions constituant des ensembles non convexes et même disjoints. L'inversion a été considérée ici comme un complément à l'optimisation robuste dans laquelle l'optimum obtenu ne satisfaisait pas une des contraintes. Les méthodes d'inversion permettent alors de résoudre ce problème en trouvant plusieurs combinaisons des entrées qui satisfont la contrainte sous la condition de rester proche de l'optimum robuste. Le but est d'atteindre une valeur cible de la contrainte non satisfaite tout en respectant les autres contraintes du système auxquelles on ajoute la condition de proximité à l'optimum.

Appliquée au dimensionnement d'un compresseur HP en avant-projets, cette méthodologie s'inscrit dans l'amélioration et l'accélération des études marquées par de nombreux rebouclages chronophages en termes de ressources informatiques et humaines.

Mots clés : Turboréacteur, dimensionnement avant-projets, résolution de problèmes inverses, problèmes mal posés, conception robuste, optimisation multi-objectifs sous contraintes, réduction de la dimension, méta-modélisation, plans d'expériences, code de calculs.

The aim of this PhD dissertation is to propose a new approach to improve and accelerate preliminary design studies for turbofan engine components. The approach consists in providing a comprehensive methodology for robust design under constraints, following three stages : dimension reduction and metamodeling, robust design under constraints, and inverse problem solving under constraints. These are the three main subjects of this PhD dissertation.

Dimension reduction is an essential pre-processing for any study. Its aim is to keep only inputs with large effects on a selected output. This selection reduces the dimension of the domain on which the study is performed. It then reduces the computational cost of the following studies and eases the understanding of the system. Metamodeling also contributes to these two objectives by replacing the time-consuming computing code by a fast metamodel apporoximating adequatly the relationship between studied output and system inputs.

Robust design under constraints is a bi-objectives optimization where different uncertainty sources are included. First, it requires to collecting and modelling uncertainties. Then a propagation method of uncertainties in the computation code has to be chosen, in order to estimate moments (mean and standard deviation) of output distribution. Optimization of these moments are the two robust design objectives. Finally, a multi-objectives optimization method must be chosen to find a robust optimum under constraints.

The development of methods to solve ill-posed inverse problems is the innovative part of this PhD dissertation. These problems can have infinitely many solutions constituting non convex or even disjoint sets. Inversion is considered here as a complement to robust design cases in which the resulting optimum doesn't satisfy one of the constraints. Inverse methods then enable to solve this problem by finding several input datasets which satisfy constraints and a condition of nearness to the optimum. The aim is to reach a target value of the unsatisfied constraint while respecting other system constraints and the condition to stay close to the optimum.

Applied to preliminary design of high pressure compressor, this methodology contributes to improvement and acceleration of studies characterizes by a lot of loopbacks time-consuming in terms of computer-based and human resources.

Keywords : Turbofan engine, preliminary design, inverse problem solving, ill-posed problems, robust design, multi-objective optimization under constraints, dimension reduction, metamodeling, design of experiments, computing code.