

## Modèles réduits à partir d'expériences numériques

**Title:** Reduced model from numerical experiments

Yves Auffray<sup>1,2</sup>, Pierre Barbillon<sup>2</sup> et Jean-Michel Marin<sup>3</sup>

**Résumé :** Les expériences numériques apparaissent dans un grand nombre de problématiques actuelles car elles permettent de remplacer des expériences physiques difficilement réalisables. Elles consistent en des évaluations d'une fonction pour des entrées fixées. Cette fonction est généralement de type boîte noire c'est-à-dire qu'elle n'est accessible qu'à travers des appels souvent coûteux à un code de calcul. Il est alors intéressant de proposer un modèle réduit de cette fonction qui pourra être appelé de manière intensive du fait du faible coût de ses évaluations. Ce modèle réduit est généralement nommé dans la littérature un méta-modèle. Dans cet article, la fonction type boîte noire considérée décrit le comportement d'un moteur d'avion. Une difficulté supplémentaire est que le domaine de définition de la fonction (c'est-à-dire l'ensemble des conditions de vol admissibles) n'est pas connu. L'objectif est d'abord de fournir une estimation de la fonction indicatrice associée au domaine de définition, puis un méta-modèle aussi fidèle et simple que possible qui approchera la fonction sur le domaine de définition. Ce travail est un préalable à une analyse de sensibilité ou à des estimations de probabilités concernant les sorties.

**Abstract:** Computer experiments are used when physical experiments are impracticable. A computer experiment consists of an evaluation, for a given input, of a function which describes the physical model. This function is said to be an expensive black-box function since it is known through a long time-consuming code only. Thus, a reduced model which approximates it and is fast to compute is needed to make analyses tractable. It is usually called a metamodel. In this study, the expensive black-box function modelizes the behaviour of an aircraft engine. Another difficulty comes from the fact that the domain of definition of the function which represents the acceptable flight conditions is unknown. Hence, we aim at providing an estimation of the indicator function associated to the domain and a metamodel which will approximate the function. This work is preliminary to a sensitivity analysis.

**Mots-clés :** expériences simulées, krigeage, classification supervisée, SVM, CART

**Keywords:** computer experiments, Kriging, classification, SVM, CART

**Classification AMS 2000 :** , ,

### 1. Introduction

Dans cet article, nous montrons sur un problème appliqué comment diverses techniques d'apprentissage statistique interviennent dans le cadre des expériences numériques. Des méthodes de classification supervisée sont utilisées afin de déterminer le domaine de définition de la fonction boîte noire représentant le modèle physique. Ensuite à partir d'un plan d'expérience numérique tiré dans le domaine estimé, un modèle réduit est construit par krigeage. Enfin, par une sélection de variables et de régresseurs, ce modèle est amélioré.

<sup>1</sup> Dassault Aviation

<sup>2</sup> Université Paris-sud 11, INRIA Saclay

E-mail : yves.auffray@math.u-psud.fr and E-mail : pierre.barbillon@math.u-psud.fr

<sup>3</sup> Institut de Mathématiques et Modélisation de Montpellier, UMR CNRS 5149, Université Montpellier 2

E-mail : jean-michel.marin@univ-montp2.fr

On s'intéresse ici à une étude d'avant-projet où l'on modélise le comportement d'un avion. Pour ce faire, on vise à obtenir un simulateur global en réunissant des simulateurs décrivant ces différentes composantes. Le comportement du moteur est décrit par un code de calcul peu maniable et d'exécution assez lente fourni par le motoriste. Il est donc assez difficilement intégrable à la modélisation globale. Notre objectif est alors de proposer un modèle réduit du moteur qui sera d'exécution quasi-instantanée, fidèle au code initial et si possible analytique.

L'ensemble des conditions de vol admissibles pour le moteur est un domaine  $E$  inconnu inclus dans un compact  $B \subset \mathbb{R}^{10}$  connu. De manière plus précise, nous disposons d'une valeur minimale et d'une valeur maximale pour chaque entrée qui permettent de définir l'ensemble  $B$  comme un hyperrectangle. Le vecteur  $\mathbf{x} = (x_1, \dots, x_{10})$  des variables d'entrée contient :

1.  $x_1$  : l'altitude (en pieds),
2.  $x_2$  : le Mach (ou nombre de Mach) qui est le rapport entre la vitesse de l'avion et la vitesse du son dans le milieu ambiant,
3.  $x_3$  : la température ambiante relative (en degré Rankine) qui est l'écart de température par rapport à la température de référence à l'altitude indiquée,
4.  $x_4$  : le pourcentage du régime moteur utilisé,
5.  $x_5$  : la perte de charge entre l'entrée du moteur et le ventilateur,
6.  $x_6$  : le prélèvement d'air au niveau du ventilateur (pourcentage du flux d'air total),
7.  $x_7$  : le prélèvement d'air basse pression sur le port numéro 1 (pourcentage du flux d'air total),
8.  $x_8$  : le prélèvement d'air basse pression sur le port numéro 2 (pourcentage du flux d'air total),
9.  $x_9$  : le prélèvement d'air haute pression total (pourcentage du flux d'air total),
10.  $x_{10}$  : le prélèvement de puissance du moteur (en chevaux vapeur sur l'arbre, shaft horsepower).

Pour une condition de vol, le code indique si elle est admissible et le cas échéant fournit un vecteur  $\mathbf{y} \in \mathbb{R}^4$  de quatre sorties d'intérêt qui sont :

1. la consommation spécifique de carburant (en livre par heure par livre force),
2. le flux d'air (en livre par seconde),
3. la poussée (en livre force),
4. la vitesse (en pied par seconde).

Nous notons  $f$  la fonction correspondant au modèle du moteur et un appel au code en une entrée  $\mathbf{x}$  admissible donne :

$$\mathbf{y} = f(\mathbf{x}).$$

Le temps de calcul de  $f$  est de l'ordre d'une dizaine de minutes pour mille points.

Avant de construire un méta-modèle, nous cherchons à obtenir de manière explicite l'ensemble des conditions de vol admissibles qui forme le domaine  $E$ . Nous traitons la détermination de ce domaine comme un problème de classification supervisée à deux classes : dans le domaine

ou hors du domaine. Afin d'apprendre la fonction indicatrice du domaine, nous utiliserons des techniques telles que les SVM (Support Vector Machines, [20]) et CART [4]. Ces méthodes sont aussi présentées dans [7, 2].

À partir de l'indicatrice estimée du domaine, nous construisons un plan d'expérience inclus dans le domaine  $E$ . Pour ce faire, des plans exploratoires sont choisis. Nous privilégierons des plans de type MAXIMIN ([8]). Le méta-modèle est obtenu en modélisant la fonction  $f$  comme la réalisation d'un processus gaussien. Cette méthode est appelée krigeage en géostatistiques et a été introduite dans le cadre des expériences simulées par [14]. La construction de plans d'expérience et de méta-modèles est décrite dans [6, 9, 15, 16].

La partie 2 présente notre étude du domaine. Il s'agit d'un mélange entre les techniques de classification et la prise en compte d'informations fournies par des experts. La partie 3 présente la construction du méta-modèle. Nous y présentons des techniques de sélection de variables et de régresseurs afin d'ajuster au mieux la qualité prédictive du méta-modèle.

## 2. Recherche du domaine

À un point  $\mathbf{x}_i \in B$  correspond une étiquette  $z_i$ . On veut estimer la fonction de classification :

$$c : \mathbf{x}_i \in B \mapsto z_i \in \{-1, +1\}.$$

Par convention,  $z_i = -1$  si  $\mathbf{x}_i \notin E$  et  $z_i = 1$  si  $\mathbf{x}_i \in E$ . Dans cette partie, les échantillons d'apprentissage et de test sont tirés dans l'hypercube  $B$  suivant une loi uniforme. Pour un échantillon donné, on appelle le code afin de déterminer les labels des points (dans ou hors du domaine). Un échantillon d'apprentissage de 1000 points, noté  $L$ , sera utilisé pour construire  $\hat{c}$  qui estimera  $c$  et un échantillon de 3000 points, noté  $U$ , tirés de manière indépendante permettra de tester les performances de la fonction de classification. Nous faisons le choix de prendre l'échantillon  $U$  de taille plus importante que l'échantillon d'apprentissage puisque cet échantillon de test est fixé au début de l'étude alors que nous nous autoriserons éventuellement à tirer de nouveaux points d'apprentissage si cela s'avère nécessaire.

### 2.1. Méthodes statistiques

Nous nous aidons de deux méthodes d'apprentissage statistique utilisées pour la classification ([7, 2]) :

- la méthode CART qui permet de créer un arbre binaire de classification de la forme

$$\hat{c}(\mathbf{x}) = \sum_{k=1}^{M_1} \mathbb{I}(\mathbf{x} \in R_k) - \sum_{l=M_1+1}^{M_1+M_2} \mathbb{I}(\mathbf{x} \in R_l),$$

où  $\{R_1, \dots, R_{M_1+M_2}\}$  constitue une partition de  $B$  et ses éléments sont des produits de fonctions indicatrices de la forme  $\mathbf{x} \mapsto \mathbb{I}(x_i < s)$  ou  $\mathbf{x} \mapsto \mathbb{I}(x_i > s)$  pour  $i \in \{1, \dots, 10\}$  et  $s \in \mathbb{R}$ . Les coupures  $\{x_i < s\}$  ou  $\{x_i > s\}$  sont choisis de manière à créer une partition qui minimise l'impureté sur le jeu d'apprentissage (mesurée ici par l'indice de Gini). Des partitions

successives sont créées jusqu'à obtenir des noeuds "purs", noeuds qui ne contiennent que des points du jeu d'apprentissage qui ont la même étiquette. Une partition intermédiaire est choisie (étape d'élagage) grâce à un critère pénalisé de sélection de modèle. Enfin, une étiquette est attribuée à chaque élément de la partition par vote majoritaire.

- la méthode SVM qui fournit une fonction dont le signe sera l'indicatrice de domaine :

$$\hat{c} = \text{signe} \left( \lambda + \sum_{l|\mathbf{x}_l \in L} \gamma_l K_{\mathbf{x}_l} \right),$$

où les  $K_{\mathbf{x}_l}$  sont des fonctions obtenues à partir d'un noyau choisi. Pour le noyau  $K$  gaussien,

$$K_{\mathbf{x}_l}(\mathbf{x}') = K(\mathbf{x}_l, \mathbf{x}') = \exp(-\theta \|\mathbf{x}_l - \mathbf{x}'\|_2^2),$$

ou pour le noyau exponentiel,

$$K_{\mathbf{x}_l}(\mathbf{x}') = K(\mathbf{x}_l, \mathbf{x}') = \exp(-\theta \|\mathbf{x}_l - \mathbf{x}'\|_1),$$

avec  $\theta > 0$ . La fonction  $\mathbf{x} \mapsto \lambda + \sum_{l|\mathbf{x}_l \in L} \gamma_l K_{\mathbf{x}_l}$  est un hyperplan dans un domaine de dimension supérieur dont les paramètres ont été calculés pour séparer au mieux les points [17]. Peu de  $\gamma_l$  sont non nuls, les  $\mathbf{x}_l$  correspondants sont appelés vecteurs de support.

Ces deux méthodes ne s'avèrent pas assez efficaces lorsqu'elles sont appliquées directement aux données. Le taux de mauvais classement est de 9% avec les SVM et de 14% avec CART. Ces taux sont calculés à partir de l'échantillon test  $U$  construit de manière indépendante. La méthode CART souffre du fait que les partitions ne font intervenir qu'une seule variable à la fois.

Nous disposons d'informations d'experts décrivant la fonction  $f$  qui nous incitent à introduire de nouvelles variables comme combinaison des variables d'entrée. Ces dernières devraient permettre une discrimination plus efficace pour CART et de proposer une classification par SVM sur un domaine de dimension réduite.

## 2.2. Adjonction de nouvelles variables

Selon des experts, la somme des prélèvements d'air haute et basse pression i.e.  $x_7, x_8, x_9$  doit être inférieure à 15% ainsi nous définissons une première variable supplémentaire :

$$x_{7:8:9} = x_7 + x_8 + x_9.$$

Nous construisons ensuite un nouvel échantillon d'apprentissage pour lequel seulement l'altitude  $x_1$  et le Mach  $x_2$  varient. L'échantillon d'apprentissage a été construit par rapport à  $x_1, x_2$  comme une grille régulière de  $40 \times 40$  points où  $x_1 \in [0, 51000]$  et  $x_2 \in [0, 0.97]$ . Les huit autres variables ont été fixées à des valeurs considérées comme standard par des experts. On espère obtenir ainsi une projection du domaine dans le plan engendré par ces deux variables. La figure 1 indique que certaines combinaisons de l'altitude et du Mach sont clairement à l'origine d'une sortie du domaine. Les points noirs correspondent à des points dans le domaine.

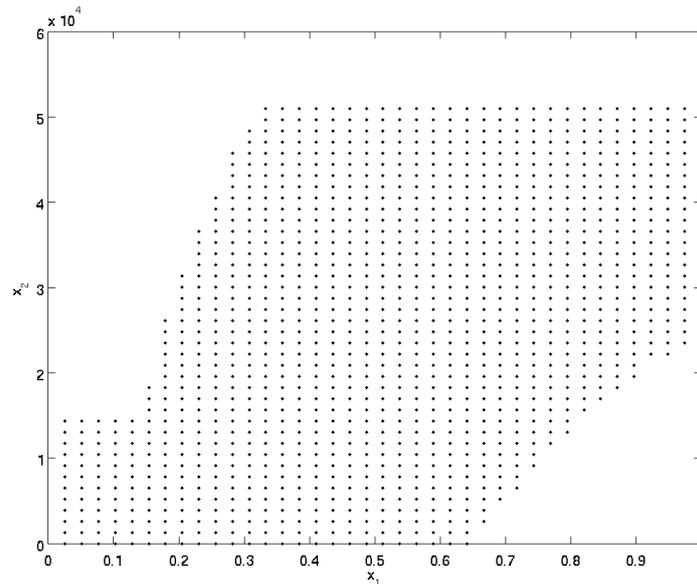


FIGURE 1. Points de vol admissibles pour une grille régulière sur l'altitude  $x_1$  et le Mach  $x_2$ .

À partir de cette constatation, des experts proposent d'ajouter comme seconde nouvelle variable la vitesse conventionnelle. Elle s'exprime en fonction de l'altitude et du Mach. Nous la notons *CAS* et elle est définie ainsi :

$$CAS = a_0 \cdot g_\gamma^{-1} \left( (1 - x_1/Z_0)^\alpha g_\gamma(x_2) \right),$$

où, pour  $x \in \mathbb{R}$ ,

$$g_\gamma(x) = \left( 1 + \frac{\gamma-1}{2} x^2 \right)^{\frac{\gamma}{\gamma-1}} - 1.$$

Ici, nous rappelons que l'altitude  $x_1$  est exprimée en pieds, et les constantes valent  $\gamma = 1.4$ ,  $T_0 = 288.15$ ,  $a_0 = \frac{340}{1852} \cdot 3600$ ,  $Z_0 = 1000 \cdot \frac{T_0}{1.9812}$ ,  $\alpha = 5.2558$ . *CAS* est exprimée en noeuds et il s'avère que la frontière observée sur la figure 1 correspond en partie à deux isocourbes :  $CAS = 70$  et  $CAS = 425$ .

De la même manière, nous étudions le domaine de définition suivant les combinaisons entre l'altitude  $x_1$  et la température ambiante. La température ambiante, notée  $x_T$ , est une modification de la température ambiante relative  $x_3$  :

$$x_T = \begin{cases} x_3 + 15 - 3.5x_1 & \text{si } x_1 \leq 36000 \text{ pieds,} \\ x_3 - 56 & \text{sinon.} \end{cases}$$

Des experts nous ont suggéré d'étudier le domaine selon la relation de l'altitude avec la température ambiante plutôt qu'avec la température ambiante relative. Nous construisons un autre échantillon d'apprentissage qui correspond à une grille  $40 \times 40$  points sur les variables  $x_1, x_T$  et à des valeurs standards fixées pour les autres variables d'entrée ( $x_2, x_4, \dots, x_{10}$ ).

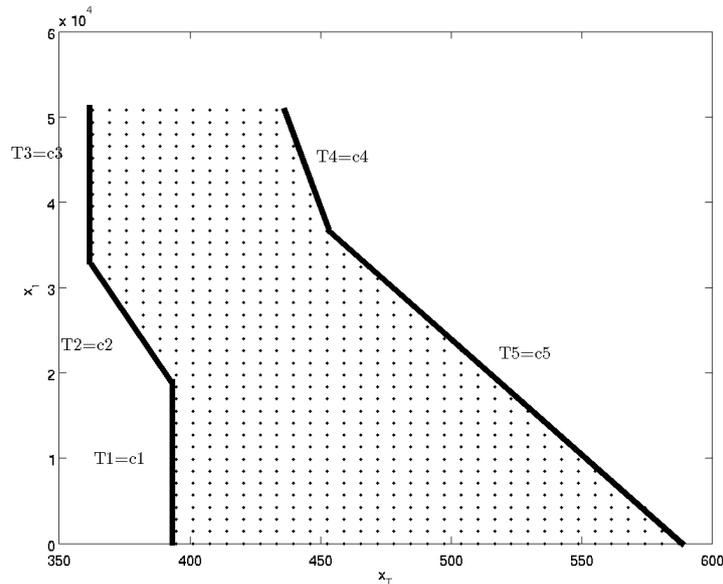


FIGURE 2. Points de vol admissibles pour une grille régulière sur l'altitude  $x_1$  et la température ambiante  $x_T$ .

Le domaine peut être délimité dans ce plan par des droites comme le montre la figure 2. On définit alors ces cinq nouvelles variables :

$$T_i = a_i x_T + b_i x_1 \quad (1 \leq i \leq 5).$$

Les constantes  $c_i$  permettent de donner les équations des droites  $T_i = c_i$ ,  $i = 1, \dots, 5$  qui forment la frontière du domaine.

Dans cette partie, nous avons introduit huit nouvelles variables :  $x_{7:8:9}$  une somme de certains prélèvements d'air, CAS la vitesse conventionnelle,  $x_T$  la température ambiante et cinq variables  $T_1, \dots, T_5$  qui sont des combinaisons linéaires de la température ambiante et de l'altitude.

### 2.3. Résultats et conclusion

La méthode CART a encouragé les ajouts de nouvelles variables en les intégrant de manière prépondérante dans la discrimination. Par exemple, à la racine, la discrimination se fait selon  $x_{7:8:9} < 15\%$ . On a aussi constaté que certaines variables n'intervenaient pas ou peu et le cas échéant, proche des feuilles de l'arbre. L. Breiman et al. [4] ont proposé une méthode dite d'importance qui donne une hiérarchie des variables explicatives. Cette méthode peut aussi être utilisée afin de tester le pouvoir discriminant des nouvelles variables.

Cette constatation incite à ne construire une classification par une méthode SVM que sur l'altitude, le mach et la température relative ambiante pour des points qui vérifient  $x_{7:8:9} < 15\%$ . C'est à dire que nous construisons une classification par SVM sur un échantillon d'apprentissage

où seules ces trois entrées varient, les autres ont été fixées à des valeurs standards (en particulier  $x_7 = x_8 = x_9 = 0$ ). Ainsi nous réduisons la dimension de l'échantillon d'apprentissage. La fonction de classification  $\hat{c}$  estimée est alors le test  $x_{7;8;9} < 15\%$ , combiné à la classification par SVM sur les trois premières variables si le test a été positif. Ceci permet d'avoir un taux d'erreur inférieur à 2%.

Si seule la méthode CART est utilisée avec toutes les variables introduites dans la partie 2.2, les taux d'erreurs obtenus sont entre 1% et 2%. Mais l'arbre de classification dépend trop fortement de l'échantillon d'apprentissage. Nous construisons alors "manuellement" l'arbre de classification. Voici les conditions :

1.  $C_1 : CAS \leq 425$  noeuds.
2.  $C_2 : CAS \geq 70$  noeuds.
3.  $C_3 : ALT \geq 15000$  pieds.
4.  $C_4 : T_1 > c_1$ .
5.  $C_5 : T_2 > c_2$ .
6.  $C_6 : T_3 > c_3$ .
7.  $C_7 : T_4 < c_4$ .
8.  $C_8 : T_5 < c_5$ .
9.  $C_9 : x_{7;8;9} < 15\%$ .

Les trois premières conditions viennent du graphique liant l'altitude avec le mach, et les cinq suivantes de celui liant la température ambiante avec l'altitude. Il suffit d'ajouter la condition sur la somme des prélèvements d'air afin d'obtenir le domaine. D'où la formule logique suivante, pour qu'un point soit dans le domaine il faut :

$$C_9 \wedge (C_1 \wedge (C_2 \vee C_3)) \wedge (C_8 \wedge (C_5 \wedge C_6 \vee \overline{C_5} \wedge C_4) \vee \overline{C_8} \wedge C_7).$$

Cette fonction de classification nous permet de prédire si un point de  $B$  est dans le domaine  $E$  avec un taux d'erreur de 0.5%. C'est la méthode retenue. Nous rappelons que tous les taux d'erreur sont calculés à partir de l'échantillon de test  $U$  défini au début de la partie 2.

Les méthodes d'apprentissage statistiques nous ont servi de guide afin de construire cet arbre de classification. Leur application directe n'a pas été concluante. Il nous a fallu construire plusieurs échantillons d'apprentissage afin de formuler des hypothèses sur le domaine et d'introduire de nouvelles variables. Nous avons utilisé un premier échantillon d'apprentissage de 1000 points, puis deux grilles de 1600 points chacune et un échantillon de test de 3000 points soit un total de 7200 appels au code pour apprendre le domaine de définition de  $f$ . Bien que les SVM ne soient finalement pas utilisées, la combinaison avec CART est une démarche intéressante car elle commence par discriminer suivant des conditions simples et permet ensuite d'appliquer une classification par SVM sur des variables dont le nombre a été réduit.

### 3. Construction d'un méta-modèle

La fonction de classification obtenue à la section précédente nous permet de construire un plan d'expérience dans le domaine  $E$ . Afin d'avoir un méta-modèle proche de  $f$  en tout point du

domaine de définition, le plan d'expérience doit être exploratoire c'est-à-dire avoir ses points bien dispersés dans ce domaine. Le critère MAXIMIN [8] garantit cette propriété et un plan d'expérience de 1300 points optimisant ce critère est construit en utilisant l'algorithme proposé par [1]. La fonction  $f$  est évaluée aux points du plan d'expérience et on obtient alors un jeu de données  $(\mathbf{x}_i, y_i = f(\mathbf{x}_i))_{1 \leq i \leq n}$  où  $\mathbf{x}_i \in \mathbb{R}^d$  où  $d = 10$ . Pour la simplicité de l'exposé, on considère que  $y_i \in \mathbb{R}$ . Cela revient à construire un méta-modèle par sortie. Le nombre de données  $n$  est légèrement inférieur à 1300 puisque le domaine n'étant pas estimé de manière exacte, certains points du plan d'expérience sont en dehors de ce dernier. Le plan d'expérience inclus dans  $E$  est noté  $D = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ .

### 3.1. Modélisation par un processus gaussien

Cette méthode vient du krigeage utilisé pour modéliser des données spatiales en géostatistiques (voir [5, 18]). Cela consiste à interpréter la fonction  $f$  comme la réalisation d'un champ gaussien  $(Y_{\mathbf{x}})_{\mathbf{x} \in E}$ . On pose le modèle suivant :

$$Y_{\mathbf{x}} = H(\mathbf{x})^T \beta + Z(\mathbf{x}) \quad (1)$$

où

- $H(\mathbf{x})$  est un vecteur de  $p$  fonctions fixées, elles forment une base d'un sous espace vectoriel noté  $\mathcal{N}$  des fonctions  $\mathbb{R}^d \rightarrow \mathbb{R}$ ,
- $\beta$  est un vecteur de  $p$  paramètres à estimer ;
- $Z$  est un processus aléatoire gaussien centré et défini par son noyau de covariance noté  $K$ .

Les évaluations de la fonction  $f$  aux points du plan d'expérience  $D$  donnent les réalisations de  $Y_{\mathbf{x}_1}, \dots, Y_{\mathbf{x}_n}$  qui sont respectivement  $y_1 = f(\mathbf{x}_1), \dots, y_n = f(\mathbf{x}_n)$ . Le méta-modèle  $\hat{f}$  est la réalisation du meilleur (au sens de l'erreur quadratique) prédicteur linéaire sans biais pour ces observations. Il s'écrit ([9]), pour  $\mathbf{x}_0 \in E$  :

$$\hat{f}(\mathbf{x}_0) = H(\mathbf{x}_0)^T \hat{\beta} + \Sigma_{\mathbf{x}_0 D}^T \Sigma_{DD}^{-1} (y_D - H \hat{\beta}),$$

où

$$\begin{aligned} y_D &= (y_1, \dots, y_n), \\ \hat{\beta} &= (H_D^T \Sigma_{DD}^{-1} H_D)^{-1} H_D^T \Sigma_{DD}^{-1} y_D, \\ H_D &= (H(\mathbf{x}_1), \dots, H(\mathbf{x}_n))^T, \\ \Sigma_{\mathbf{x}_0 D} &= (K(\mathbf{x}_1, \mathbf{x}_0), \dots, K(\mathbf{x}_n, \mathbf{x}_0))^T, \\ \Sigma_{DD} &= (K(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i, j \leq n}. \end{aligned}$$

Le vecteur  $\hat{\beta}$  correspond à l'estimation par maximum de vraisemblance du paramètre  $\beta$  sous les hypothèses du modèle (1). Le méta-modèle obtenu est la somme d'une partie régression  $H(\mathbf{x}_0)^T \hat{\beta}$  et d'une interpolation sur les résidus de la régression  $\Sigma_{\mathbf{x}_0 D}^T \Sigma_{DD}^{-1} (y_D - H \hat{\beta})$ . Nous remarquons qu'en un point du plan d'expérience i.e. si  $\mathbf{x}_0 = \mathbf{x}_i$ , on a

$$\hat{f}(\mathbf{x}_i) = f(\mathbf{x}_i).$$

Cette propriété d'interpolation des points du plan d'expérience est intéressante car elle permet d'assurer que pour des points de vol "standards", la prédiction est exacte à la seule condition que ces points fassent partie du plan d'expérience.

### 3.2. Régresseurs et noyaux utilisés

Nous utilisons principalement les régresseurs et les noyaux proposés par le programme DACE ([10]) et avons ajouté les noyaux décrits dans [9]. Les espaces de régresseurs considérés sont :

$$\mathcal{N}_0 \subset \mathcal{N}_1 \subset \mathcal{N}_2 \subset \mathcal{N}_3,$$

où  $\mathcal{N}_l = \{\prod_{j=1}^d x_j^{v_j} \mid 0 \leq v_1, \dots, v_n \leq l, \sum_{i=1}^n v_i \leq l\}$ . On se limitera aux polynômes de degré trois car la famille  $\mathcal{N}_l$  comporte en dimension  $d$ ,  $p = C_{l+n}^l$  régresseurs. Dans notre cas,  $C_{3+10}^3 = 286$ .

On fait l'hypothèse de stationnarité au second ordre sur le processus  $Z$ , i.e pour  $\forall \mathbf{x}, \mathbf{x}' \in E$ ,

$$\text{Cov}(Z(\mathbf{x}), Z(\mathbf{x}')) = \sigma^2 K(\mathbf{x}, \mathbf{x}') = \sigma^2 R(\mathbf{e}),$$

ne dépend que du vecteur  $\mathbf{e}$  des écarts suivant chaque dimension. On a noté  $\mathbf{e} = (e_j = |x_j - x'_j|)_{j=1, \dots, d}$ . On impose  $R(0) = 1$  ainsi le noyau décrit la corrélation et  $\sigma^2 > 0$  est un paramètre de variance. Le noyau  $K$  est de plus défini comme une tensorisation de noyaux univariés qui sont des fonctions radiales de base :

$$R(\boldsymbol{\theta}, \mathbf{e}) = \prod_{j=1}^d r((\theta_j, (\theta)_{j>d}), e_j).$$

Les paramètres  $\theta_j$  pour  $1 \leq j \leq d$  sont des paramètres d'échelle. On peut choisir de les prendre tous égaux pour avoir un modèle isotrope, mais nous utilisons plutôt un modèle anisotrope. Les paramètres (optionnels)  $\theta_j$  avec  $j > n$  sont des paramètres spécifiques au noyau univarié  $r$ . Les paramètres  $\theta$  et  $\sigma^2$  sont estimés par maximum de vraisemblance ([16]). Nous noterons  $\hat{\theta}$  et  $\hat{\sigma}^2$  leurs estimations. Les noyaux univariés comparés dans notre étude sont

*De type exponentiel* :  $r((\theta_j, (\theta)_{j>d}), e_j) = \exp(-\theta_j |e_j|^{\theta_{d+1}})$ , pour  $0 < \theta_{d+1} \leq 2$ . Si on fixe  $\theta_{d+1} = 1$ , le noyau est dit exponentiel (exp), si on fixe  $\theta_{d+1} = 2$  il est dit gaussien, sinon il est dit exponentiel généralisé (expg). Il n'est différentiable en moyenne quadratique que dans le cas gaussien, ce qui correspond à un processus dont les réalisations sont très lisses.

*Linéaire* :  $r((\theta_j, e_j) = \max\{0, 1 - \theta_j |e_j|\}$ .

*Cubique* :  $r((\theta_j, e_j) = 1 - \frac{3(1-\rho)}{2+\gamma} \xi_j^2 + \frac{(1-\rho)(1-\gamma)}{2+\gamma} \xi_j^3$  pour  $\xi_j = \min\{1, \theta_j |e_j|\}$ . Il doit vérifier la condition

$$\rho \geq \frac{5\gamma^2 + 8\gamma - 1}{\gamma^2 + 4\gamma + 7}$$

et  $\rho \in [0, 1]$ ,  $\gamma \in [0, 1]$  afin d'être défini positif (voir [12]). Les processus définis ainsi sont différentiable une fois en moyenne quadratique. Le programme DACE propose seulement le noyau "spherical" qui correspond à  $\rho = \gamma = 0$ .

*Matérn* :  $r((\theta_j, (\theta)_{j>d}), e_j) = \frac{(\theta_j |e_j|)^{\theta_{d+1}}}{\Gamma(\theta_{d+1}) 2^{\theta_{d+1}-1}} J_{\theta_{d+1}}(\theta_j |e_j|)$  où on doit avoir  $\theta_j \in (0, \infty)$  et  $\theta_{d+1} \in (-1, \infty)$ .  $J_{\theta_{d+1}}$  est une fonction de Bessel modifiée d'ordre  $\theta_{d+1}$ . Le processus associé sera  $m$  fois différentiable en moyenne quadratique si et seulement si  $\theta_{d+1} > m$ . Ainsi la régularité du processus est gouverné par le paramètre  $\theta_{d+1}$  tandis que  $\theta_j$  contrôle l'échelle de corrélation.

### 3.3. Méthodes

Nous présentons les méthodes de test de modèles puis les pistes que nous avons explorées afin d'améliorer le méta-modèle obtenu par krigeage. Nous ne proposons pas un algorithme permettant un traitement systématique des données comme dans [11]. Nous privilégions une méthode pilotée par l'utilisateur.

**Qualité d'un modèle** Afin de juger les estimateurs obtenus, nous avons besoin de critères relatifs et absolus. Pour comparer deux estimateurs, nous calculons pour chacun d'entre eux l'erreur quadratique sur un échantillon de test  $U$  d'environ 2000 points tirés de manière uniforme et indépendante dans  $E$  et indépendant de l'échantillon d'apprentissage sur lequel le méta-modèle  $\hat{f}$  a été construit :

$$EQ = \frac{1}{|U|} \sum_{\mathbf{x} \in U} (f(\mathbf{x}) - \hat{f}(\mathbf{x}))^2. \quad (2)$$

Quant à la qualité d'un estimateur, elle est appréhendée par les erreurs relatives :

$$\left| \frac{f(x) - \hat{f}(x)}{f(x)} \right|, \quad (3)$$

dont on calcule la moyenne et le quantile à 95%.

**Sélection de variables** Étant donné que toutes les variables d'entrée ne sont pas forcément utiles pour calculer une sortie, nous proposons d'effectuer une sélection de variables. Une procédure exhaustive qui testerait toutes les combinaisons possibles des dix variables seraient trop coûteuse. Nous utiliserons alors les méthodes de sélection ascendante et descendante. Les modèles construits à partir de différentes combinaisons de variables seront comparés sur la base de l'erreur quadratique (2) et des vraisemblances pénalisées AIC et BIC :

$$AIC = \mathcal{L}(\hat{\theta}, \hat{\sigma}^2, \hat{\beta}, y_D) - \text{Nbparametres},$$

$$BIC = \mathcal{L}(\hat{\theta}, \hat{\sigma}^2, \hat{\beta}, y_D) - \frac{1}{2} \log(n) \cdot \text{Nbparametres},$$

$$\text{avec } \mathcal{L} = -\frac{n}{2} \log(2\pi\hat{\sigma}^2) - \frac{1}{2} \log(|\Sigma_{DD}(\hat{\theta})|) - \frac{(y_D - H_D \hat{\beta})^T \Sigma_{DD}(\hat{\theta})^{-1} (y_D - H_D \hat{\beta})}{2\hat{\sigma}^2},$$

où le nombre de paramètres est le nombre de coordonnées de  $\theta$  auquel on ajoute le nombre de coordonnées de  $\beta$  et 1 pour  $\sigma^2$ . Ces deux critères sont à maximiser tandis que l'erreur quadratique est un critère à minimiser. Contrairement à l'erreur quadratique, les critères AIC et BIC ont l'avantage de ne pas nécessiter l'utilisation d'un jeu de test (indépendant des données d'apprentissage).

**Méthode descendante** : Nous commençons par considérer le modèle complet (construit à partir des  $d$  variables d'entrée) et cherchons quel sous-modèle construit à partir de  $d - 1$  variables est optimal au sens du critère retenu (AIC, BIC ou EQ). Nous conservons alors ce sous-modèle ce qui revient à éliminer une variable d'entrée. Puis, nous itérons cette procédure jusqu'à obtenir un modèle contenant une seule variable.

*Méthode ascendante* : Nous commençons par chercher quel modèle construit à partir d'une variable, est optimal au sens du critère retenu. Ensuite, nous ajoutons la variable qui ne fait pas encore partie du modèle et qui donne, une fois ajoutée, la meilleure valeur du critère. Nous continuons à enrichir ainsi le modèle jusqu'à obtenir le modèle complet.

Finalement, chaque méthode conduit à une suite de modèles emboîtés. Pour chaque méthode, le modèle retenu est celui qui est le meilleur de la suite au sens du critère de sélection. Au cas où elles n'aboutissent pas au même modèle, l'on conserve bien évidemment le meilleur des deux au sens du critère.

**Sélection de régresseurs** Afin de choisir les régresseurs dans le modèle (1), nous proposons une méthode de seuillage. Pour un modèle donné, nous calculons l'estimation  $\hat{\beta}$  et nous éliminons les régresseurs correspondant à une composante de  $\hat{\beta}$  inférieure à un seuil fixé. Cela donne un nouveau modèle sur lequel on "ré-estime" les paramètres. Ce procédé est employé pour différentes valeurs de seuils. Nous conservons en définitive le modèle pour lequel l'erreur quadratique calculée est la plus faible. Nous pourrions ainsi inclure des monômes de degré trois sans pour autant construire un modèle avec tous les monômes de degré trois (en nombre trop important en dimension 10). Cette méthode est simple à mettre en œuvre. Des méthodes de régularisation telles que LASSO [19] peuvent aussi être utilisées.

**Régularisation** Le programme de krigeage utilisé, DACE, construit tous ses prédicteurs en considérant que les observations  $y_i$  sont perturbées par un bruit blanc  $\varepsilon$  de variance  $\omega^2$  i.e.

$$y_i = f(x_i) + \varepsilon_i.$$

La variance  $\omega^2$  est de l'ordre de  $n \cdot 2^{-52}$ . C'est avant tout une technique numérique qui vise à empêcher un mauvais conditionnement de la matrice de corrélation  $\Sigma_{DD}$  entre les points du plan d'expérience. Cette méthode est une régularisation du prédicteur (similaire au cas des fonctions splines [21]).

Nous commencerons par tester les différentes combinaisons de familles de régresseurs et de noyaux présentés dans la partie 3.2. Ensuite, nous appliquerons les méthodes de sélection de variables et enfin nous choisirons les régresseurs dans la famille retenue. Nous emploierons la régularisation que propose DACE par défaut.

### 3.4. Résultats

Les résultats présentés portent sur les méta-modèles construits pour la sortie correspondant au flux d'air.

D'après la comparaison des erreurs quadratiques présentées dans le tableau 1, les noyaux type exponentiel généralisé (expg) et Matérn combinés à des régresseurs qui sont des polynômes de degré inférieur à deux ou trois fournissent les meilleurs méta-modèles. Le noyau de Matérn donne lieu à des méta-modèles qui ont les meilleures performances mais qui ne sont pas sous forme explicite ce qui peut induire une lourdeur d'utilisation. C'est pourquoi, nous étudions également les méta-modèles faisant intervenir des noyaux expg. La bonne performance des noyaux de

	gaussien	exp	lin	spherical	expg	matérn
$\mathcal{N}_0$	87.8	24.4	75.0	275.9	28.5	0.75
$\mathcal{N}_1$	6.1	4.3	5.9	13.7	3.1	0.64
$\mathcal{N}_2$	2.0	1.9	1.8	2.5	1.3	0.55
$\mathcal{N}_3$	1.3	1.2	1.2	1.5	0.92	0.68

TABLEAU 1. Comparaison des EQs des méta-modèles suivant les régresseurs et les noyaux de covariance choisis

Matérn est due à leur grande adaptativité. En effet, les paramètres décrivant un noyau de Matérn permettent de régler leur régularité.

régresseurs dans noyaux	$\mathcal{N}_3$	$\mathcal{N}_3$	$\mathcal{N}_3$	$\mathcal{N}_3$
	Matérn	expg	Matérn	expg
nombre de variables	10	10	8	8
nombre de régresseurs	50	50	98	71
EQ	0.49	0.75	0.57	0.65
Erreur relative moyenne	0.4%	0.5%	0.5%	0.5%
quantile 95%	1.2%	1.6%	1.3%	1.4%

TABLEAU 2. Performances des méta-modèles après sélection des régresseurs

Les méthodes de sélection de variables présentées dans la partie précédente, sont mises en oeuvre. Pour les méta-modèles construits avec un noyau expg, le modèle retenu a 8 variables : toutes sauf  $x_6$ , le prélèvement d'air au niveau du ventilateur, et  $x_{10}$ , le prélèvement de puissance du moteur. Par contre, pour les méta-modèles construits avec un noyau de Matérn, aucune variable d'entrée n'est éliminée. Dans ces deux cas, les deux méthodes de sélection ont conduit au même modèle pour chacun des trois critères.

Nous testons ensuite les modèles à 8 (sans  $x_6$  et  $x_{10}$ ) et 10 variables d'entrée pour les noyaux de Matérn et expg avec des régresseurs qui sont des polynômes de degré inférieur à trois. Nous appliquons une méthode de sélection des régresseurs par seuillage. Plusieurs seuils sont essayés et nous présentons dans le tableau 2, les méta-modèles qui sont optimaux au sens de la minimisation de l'erreur quadratique. Leurs performances sont comparées en terme d'erreur quadratique, de la moyenne de l'erreur relative (3) et son quantile à 95%. Le meilleur modèle est celui construit à partir du noyau de Matérn et des dix variables d'entrée. Le meilleur modèle obtenu avec un noyau expg est construit à partir de huit variables d'entrée.

Nous présentons également les performances en tant que méta-modèle de la partie régresseur seule  $\mathbf{x}_0 \in E \mapsto H(\mathbf{x}_0)^T \hat{\beta}$ . En suivant les étapes de sélection de variables et de régresseurs, on obtient une somme de 50 monômes de degré inférieur à trois ne faisant intervenir que les huit variables retenus précédemment. Ce méta-modèle a l'avantage de permettre une étude analytique simple. Ses performances sont données dans le tableau 3.

Le plan d'expérience pour construire les méta-modèles a demandé 1300 appels au code et l'échantillon  $U$  pour tester leurs performances a été tiré de manière uniforme dans  $E$  grâce à 2000 appels. Ces appels sont indépendants de ceux de la partie 2.

<b>nombre de variables</b>	8
<b>nombre de régresseurs</b>	50
<b>EQ</b>	2.1
<b>Erreur relative moyenne</b>	1%
<b>quantile 95%</b>	3.2%

TABLEAU 3. Performances du méta-modèle ne comportant que les régresseurs

#### 4. Conclusion

Au total, nous avons utilisé 10500 appels au code afin d'apprendre le domaine  $E$ , construire un méta-modèle tout en testant ses performances. Ici, le code n'était pas trop coûteux (dizaine de minutes pour mille appels), nous n'avons pas recyclé les appels utilisés lors de la recherche du domaine pour construire le méta-modèle. Cela peut servir notamment pour constituer l'échantillon de test de la qualité d'approximation. De plus, afin d'assurer une bonne répartition des points, des suites quasi uniformes telles que les suites de Sobol' [3] auraient pu à être utilisées pour construire les échantillons d'apprentissage et de test.

Dans la partie 2, envisager la recherche du domaine comme un problème de classification supervisée à deux classes a permis d'employer des techniques d'apprentissage comme CART ou les SVM. Leur utilisation sur le domaine des entrées à 10 dimensions n'a pas fourni une estimation suffisamment précise de la fonction indicatrice associée au domaine. Aussi, nous avons eu recours à des informations d'experts dont nous avons testé la pertinence par ces méthodes. L'ajout de variables, fonctions des variables d'entrée initiales, a permis une meilleure discrimination par CART. L'application des SVM sur un espace de dimension réduite après une étape de l'arbre CART s'est avérée efficace. Au final, nous avons proposé un arbre de classification binaire comme estimation de la fonction indicatrice du domaine. Toutefois, la combinaison proposée de CART et des SVM reste une stratégie intéressante dans le cas où moins d'informations sur le domaine sont disponibles.

Dans la partie 3, nous nous sommes intéressés à la construction d'un méta-modèle par une modélisation de la fonction boîte noire comme la réalisation d'un processus gaussien. Nous avons proposé des méthodes de sélection de variable et de régresseurs pour construire un méta-modèle plus précis. Le meilleur méta-modèle est obtenu en utilisant des noyaux de Matérn ce qui induit qu'il n'a pas une expression en terme de fonctions usuelles. Il est aussi possible de construire un méta-modèle ayant des performances proches à partir d'un noyau type exponentiel généralisé. Nous avons également présenté les performances d'un méta-modèle qui ne contient que la partie régresseurs. Ainsi nous obtenons trois méta-modèles pour lesquels la complexité est liée à la performance. Ce sont aux experts responsables de la modélisation globale de décider de convenir d'un compromis suivant le but de leur étude.

Le méta-modèle retenu permet alors d'envisager une analyse de sensibilité ([11, 13]) puisqu'il est d'évaluation quasi instantanée.

#### Références

- [1] Y. AUFRAY, P. BARBILLON et J.-M. MARIN : Maximin design on non-hypercube domain and kernel interpolation. Rapport technique, arXiv, 2010. <http://arxiv.org/abs/1004.0784>.

- [2] C. M. BISHOP : *Pattern Recognition and Machine Learning*. Springer, New-York, 2006.
- [3] P. BRATLEY et B. L. FOX : Algorithm 659 : Implementing sobol's quasirandom sequence generator. *ACM Trans. Math. Softw.*, 14(1):88–100, 1988.
- [4] L. BREIMAN, J. FRIEDMAN, R. OLSHEN et C. STONE : *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984.
- [5] N. CRESSIE : *Statistics for Spatial Data*. Wiley, New York, 1993.
- [6] K.-T. FANG, R. LI et A. SUDJIANTO : *Design and Modeling for Computer Experiments*. Computer Science and Data Analysis. Chapman & Hall/CRC, 2005.
- [7] T. HASTIE, R. TIBSHIRANI et J. FRIEDMAN : *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, 2001.
- [8] M. E. JOHNSON, L. M. MOORE et D. YLVISAKER : Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131 – 148, 1990.
- [9] J. R. KOEHLER et A. B. OWEN : Computer experiments. In *Design and analysis of experiments*, volume 13 de *Handbook of Statistics*, pages 261–308. North Holland, Amsterdam, 1996.
- [10] N.S. LOPHAVEN, H.B. NIELSEN et J. SONDERGAARD : Dace, a matlab kriging toolbox. Rapport technique IMM-TR-2002-12, DTU, 2002. Available to : <http://www2.imm.dtu.dk/hbn/dace/dace.pdf>.
- [11] A. MARREL, B. IOOSS, F. Van DORPE et E. VOLKOVA : An efficient methodology for modeling complex computer codes with gaussian processes. *Computational Statistics & Data Analysis*, 52(10):4731–4744, 2008.
- [12] T. MITCHELL, M. MORRIS et D. YLVISAKER : Existence of smoothed stationary processes on an interval. *Stochastic Processes and their Applications*, 35(1):109 – 119, 1990.
- [13] J. E. OAKLEY et A. O'HAGAN : Probabilistic sensitivity analysis of complex models : A bayesian approach. *Journal of the Royal Statistical Society, Series B*, 66:751–769, 2002.
- [14] J. SACKS, S. B. SCHILLER, T. J. MITCHELL et H. P. WYNN : Design and analysis of computer experiments (with discussion). *Statistical Science*, 4:409–435, 1989.
- [15] J. SACKS, S. B. SCHILLER et W. J. WELCH : Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.
- [16] T. J. SANTNER, Williams B. et Notz W. : *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.
- [17] B. SCHÖLKOPF et A. J. SMOLA : *Learning with Kernels : Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [18] M. L. STEIN : *Interpolation of Spatial Data : Some Theory for Kriging*. Springer, New York, 1999.
- [19] R. TIBSHIRANI : Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.
- [20] V. N. VAPNIK : *The Nature of Statistical Learning Theory*. Springer-Verlag, New-York, 1996.
- [21] G. WAHBA : *Spline models for observational data*, volume 59 de *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.