Contributions statistiques au calage et à la validation des codes de calcul

Guillaume Damblin

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

AgroParisTech
INSTITUT DES SCIENCES ET INDUSTRIES DU VIVANT ET DE L'ENVIRONNEMENT
PARIS INSTITUTE OF TECHNOLOGY FOR LIFE, FOOD AND ENVIRONMENTAL SCIENCES

Ecole Doctorale ABIES

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

L'Institut des Sciences et Industries du Vivant et de l'Environnement

Spécialité "Mathématiques Appliquées"

présentée et soutenue publiquement par

Guillaume DAMBLIN

le 20 novembre 2015

Contributions statistiques au calage et à la validation des codes de calcul

Jury

M. Gilles Celeux,	Directeur de recherche, INRIA & Université Paris Sud	Président
M. David Ginsbourger,	Senior Researcher, IDIAP & Université de Berne	Rapporteur
Mme. Céline Helbert,	Maître de conférences, École Centrale de Lyon	Examinatrice
M. Marc Sancandi,	Dr, Ingénieur chercheur CEA Cesta	Examinateur
M. Eric Parent,	Professeur, IGPEF, UMR MIA-518, AgroParisTech/INRA	Directeur de thèse
M. Pierre Barbillon,	Maître de conférences, UMR MIA-518, AgroParisTech/INRA	Encadrant
M. Merlin Keller,	Dr, Ingénieur chercheur EDF R&D Chatou	Encadrant
M. Alberto Pasanisi,	HdR, European Institute For Energy Research, Karlsruhe	Encadrant

AgroParisTech
16 rue Claude Bernard, 75 005 Paris
UMR MIA-518 AgroParisTech/INRA, France

Remerciements

En premier lieu, je remercie grandement Eric Parent pour avoir permis la réalisation de ce projet de thèse. Au delà de son rôle de directeur, il n'a eu de cesse de m'encourager tout au long de ces trois ans, m'octroyant la marge de manœuvre dont j'avais besoin pour m'épanouir dans mes recherches. Je garde bien-sûr à l'esprit ses raisonnements subtils sur la statistique bayésienne qui m'ont bien souvent laissé cogiter. Je ne saurais oublier les fameuses rencontres au sommet de Rochebrune où ski et statistiques font plus que bon ménage !

J'aimerais ensuite remercier très chaleureusement Pierre Barbillon pour sa grande disponibilité depuis les prémices jusqu'à l'aboutissement de ce projet. Sans nul doute, je lui dois beaucoup, tant au niveau scientifique concernant de nombreux aspects du manuscrit, que pour ses qualités humaines.

A EDF, je tiens à remercier Merlin Keller pour avoir encadré au plus près ce travail de thèse. Je lui suis très reconnaissant d'avoir répondu à nombre de mes questions de bayésien novice et de m'avoir encouragé sans relâche à valoriser mon travail en interne.

Enfin, pour en terminer avec ma garde rapprochée, je remercie celui qui fut l'instigateur de la thèse et le superviseur de ces travaux à EDF pendant les deux premières années. Sa rigueur m'a beaucoup inspiré tout comme sa capacité à se positionner sur tous les fronts de la recherche appliquée. Merci Alberto Pasanisi.

C'est avec grand honneur que je remercie maintenant Gilles Celeux et David Ginsbourger pour avoir accepté de juger ce mémoire de thèse. Leurs analyses critiques seront à n'en pas douter à l'origine de futurs travaux qui me permettront de faire fructifier encore davantage les contributions du mémoire.

Pour avoir accepté d'examiner ce travail, j'aimerais remercier Céline Helbert que j'ai déjà eu l'occasion de côtoyer à plusieurs reprises lors de congrès, et Marc Sancandi qui vient compléter ce jury et dont j'attends avec impatience de faire la connaissance. J'étudierai avec grand intérêt leurs commentaires respectifs.

Pendant ces trois années, j'ai essentiellement travaillé au centre EDF R&D de Chatou, découvert un peu plus tôt lors de mon stage de Master. A ce propos, je remercie profondément mon tuteur de l'époque Bertrand Iooss de m'avoir initié avec passion au domaine des plans d'expériences numériques. Je lui suis par ailleurs infiniment reconnaissant de m'avoir accordé sa confiance bien au delà du stage. A EDF, je remercie également l'ensemble des personnels des groupes T55 et T57 du département MRI, en particulier Emmanuel Remy grâce à qui l'on se sent forcément moins seul lorsque les journées durent (où sur la piste de danse...), mais aussi Mathieu Couplet dont la curiosité scientifique m'a impressionné.

Je tiens ensuite à remercier tous les chercheurs et doctorants de l'UMR 518 de l'AgroParisTech qui m'ont permis d'évoluer dans un laboratoire particulièrement stimulant et complémentaire de mon environnement à EDF. J'ai également une pensée pour le groupement de recherche MASCOT-NUM et le consortium ReDICE au sein desquels les nom-

breux échanges autours des thématiques de quantification d'incertitude en simulation numérique furent très intéressants. Enfin, je remercie le laboratoire de Génie Logiciel et de Simulation (LGLS) du CEA Saclay pour m'avoir permis de terminer la rédaction de ce mémoire dans les meilleures conditions, avant la suite que j'espère fructueuse.

J'adresse maintenant mes salutations au Docteur Jean-Baptiste, mon collègue de bureau à EDF pendant deux ans et demi, et aux Docteurs Pierre, François, Jeanne et Vincent à qui je souhaite le meilleur, tout comme à ceux qui se lancent dans cette aventure passionnante, Mathieu, Xavier, Joseph, Thomas et Nazih.

Ces dernières lignes sont destinées à mes amis et ma famille. De Paris à Lille en passant par Kiev (sacré détour quand même me direz vous...), j'aimerais remercier le groupe Clichy, Pierre, Kateryna, Jérôme, Lucie, Docteur Hubert, So pour tous les bons moments passés ensemble. Vous m'avez permis de laisser la thèse de côté quand il le fallait ! Enfin, je remercie très chaleureusement toute ma famille, auprès de laquelle j'ai pu trouver toute l'énergie nécessaire pour aller au bout de cette aventure.

Table des matières

Liste des figures

Chapitre 1

Introduction

Contexte

La complexité des systèmes réels étudiés aussi bien dans les études industrielles que dans les domaines de la physique, en chimie ou encore en biologie, a conduit à l'essor de la simulation numérique comme complément, voire comme un outil de substitution à l'expérience physique. En effet, celle-ci se révèle souvent coûteuse à mettre en œuvre ou impossible à effectuer, voire même inadaptée lorsque l'on s'intéresse à l'évolution d'un système réel au cours d'une échelle de temps pouvant atteindre plusieurs décennies, par exemple dans le cadre des études de sureté nucléaire. La simulation numérique consiste en l'exécution d'un programme informatique que l'on peut assimiler à une fonction réelle $y := y_{\boldsymbol{\theta}}(\mathbf{x})$ dont l'objectif est de reproduire le comportement d'un système physique d'intérêt $r(\mathbf{x})$. Le vecteur $\mathbf{x} \in \mathscr{X}$ se compose de variables dites de contrôle, que le scientifique peut mesurer, et éventuellement de variables environnementales non-observées et donc supposées aléatoires (Santner et al., 2003). Le vecteur $\boldsymbol{\theta}$ est un vecteur de paramètres spécifique au code. Par exemple, la puissance électrique nécessaire pour garder des conditions de conforts à l'intérieur d'un bâtiment peut être prédite par la réponse d'un code de calcul fonction d'une température de consigne, des conditions météo, de l'architecture du bâtiment, qui constituent le vecteur $\mathbf{x}$, mais aussi de coefficients de convection associés aux systèmes de régulation d'air à l'intérieur du bâtiment, de coefficients quantifiant les déperditions thermiques, etc, qui composent le paramètre $\boldsymbol{\theta}$.

Dans ce mémoire, on utilisera le terme de code de calcul pour désigner un programme informatique, aussi appelé modèle numérique ou simulateur dans la littérature. Au sein d'EDF, plusieurs plateformes logicielles, dont *Code_Saturne* (`http://www.code-saturne.org`) ou *Code_Aster* (`http://www.code-aster.org`), constituent respectivement les supports principaux des études en mécanique des fluides et en thermomécanique des structures. Leur construction requiert la traduction des mécanismes physiques mis en jeux sous forme d'équations mathématiques basées sur des lois de comportement et de conservation. Par exemple, en mécanique des fluides, les équations de Navier Stokes sur lesquelles repose l'implémentation du logiciel Code_Saturne traduisent le mouvement des fluides newtonien. En électromagnétisme, les équations de Maxwell décrivent le comportement des champs électriques et magnétiques. Ces systèmes d'équations aux dérivées partielles sont non linéaires et leur résolution fait appel à des méthodes numériques destinées au calcul d'une solution approchée sur un maillage de l'espace des variables. L'élaboration d'un code de calcul nécessite alors trois étapes :

1. la traduction mathématique du comportement physique du phénomène étudié sous la forme d'un système d'équations incluant des conditions aux limites,

2. l'utilisation de méthodes numériques pour résoudre des versions discrétisées du système d'équations initial sur un maillage (éléments finis, volume finis),
3. l'implémentation des algorithmes dans un langage de programmation informatique (Fortran, C, Python, ...).

Le présent mémoire s'inscrit au sein de la thématique générale de la quantification des incertitudes relative à l'utilisation de la simulation numérique dans les études industrielles. Depuis la phase d'élaboration du code jusqu'au traitement des simulations, il est indispensable de déterminer toutes les sources d'erreurs susceptibles d'affecter la capacité de celui-ci à prédire le système physique d'intérêt. La quantification des erreurs induites par la résolution approchée du problème mathématique ainsi qu'à son implémentation est appelée *Vérification*. Comme l'explique Trucano et al. (2006), l'étape de vérification d'un code nécessite une compréhension précise de son processus de fabrication, avec comme point de départ, le choix des conditions limites, du maillage, en passant par les algorithmes numériques utilisés pour la résolution du problème mathématique. Une fois cette étape correctement effectuée, il est légitime de s'interroger sur le bien-fondé de la modélisation mathématique du système physique qui est à l'origine de l'élaboration du code de calcul. On traite alors de sa *Validation*, qui motive notre travail de thèse. Les premières définitions attribuées à ce concept suggèrent une réponse binaire par oui ou par non. Mais alors, on pourrait rapidement argumenter que toute modélisation mathématique est imparfaite et que par conséquent aucun code ne peut être considéré valide. Cette première définition n'est en fait pas mise en perspective des vrais intérêts de la validation, à savoir déterminer à quel point le code de calcul prédit avec une précision suffisante le système physique d'intérêt avec en ligne de mire les enjeux d'une étude industrielle.

En s'inspirant des définitions proposées dans la littérature, la Vérification et la Validation, couramment appelés dans la communauté scientifique V&V, consistent en une procédure que l'on peut découper en trois étapes successives ; d'abord une étape dédiée à la vérification (Roache, 1998), puis deux étapes de validation :

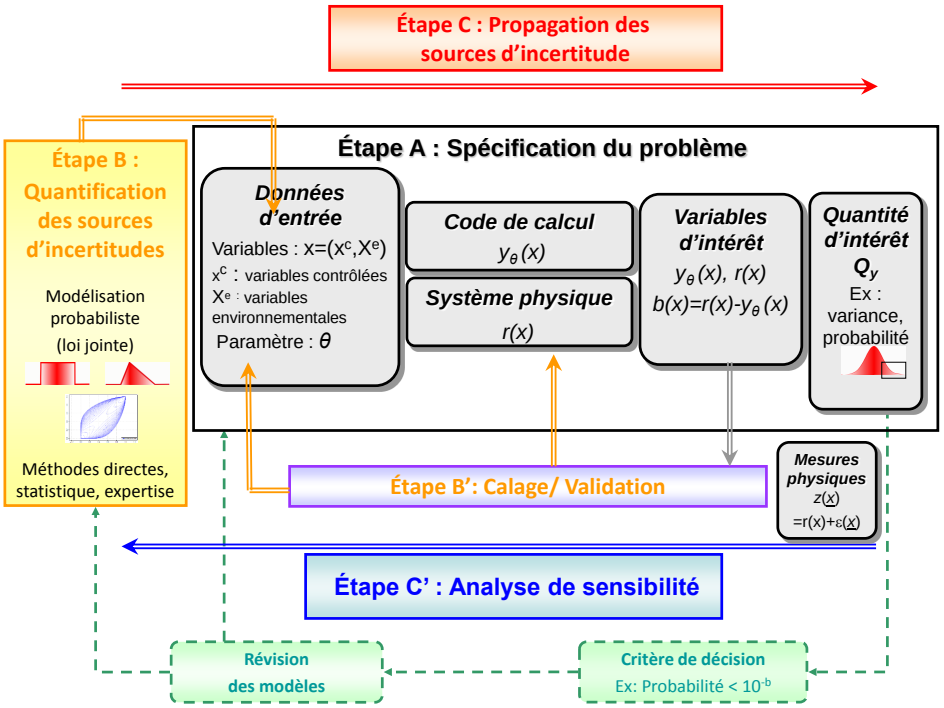
1. la première qui consiste à calculer l'incertitude de prédiction du système physique en utilisant les réponses d'un code de calcul l'approchant. Bayarri et al. (2007) parlent alors d'une approche prédictive,
2. la seconde qui doit juger si les prédictions calculées à l'étape 1 sont suffisamment précises pour répondre à une problématique industrielle aux enjeux bien identifiés (AIAA, 1998; ASME, 2009).

Par exemple, en thermique du bâtiment, on pourra s'intéresser à quantifier l'incertitude affectant la puissance électrique injectée à l'intérieur d'un bâtiment pour le chauffer sur une période de temps donnée au regard d'un objectif de prédiction de la consommation moyenne (voir chapitre 5).

Une fois cette procédure de validation effectuée, le code de calcul pourra être utilisé pour estimer la probabilité qu'une ou plusieurs quantités d'intérêt du système physique dépassent un seuil critique, pour mener une analyse de sensibilité ou encore pour répondre à une problématique décisionnelle comme par exemple la garantie de performance énergétique dans le cadre de notre application en thermique du bâtiment (voir chapitre 5). Ces différents objectifs sont ordonnés à l'intérieur d'une méthodologie statistique maintenant bien balisée de traitement des incertitudes en simulation numérique (voir figure 1.1) et (`http://www.openturns.org`). Nous rappelons les étapes successives de la méthodologie afin de mieux situer l'étape de V&V

- **l'étape A** est consacrée à la spécification du problème. Les quantités d'intérêts Q_y du code sont identifiées, par exemple la moyenne de $y_{\boldsymbol{\theta}}(\mathbf{x})$, sa variance, une pro-

FIGURE 1.1 – *Méthodologie de traitement des incertitudes en simulation numérique incluant l'étape B' de calage/validation d'un code de calcul (http://www-mip.onera.fr/projets/JSO-2012/fichiers/presentations/25janvier/B_Iooss.pdf).*



babilité de dépassement de seuil, ou simplement les réponses du code $y_{\boldsymbol{\theta}}(x)$ à l'intérieur d'un domaine de fonctionnement,

- **l'étape B** est l'étape d'identification des sources d'incertitudes pour chacune des entrées du code. Certaines composantes du vecteur $\mathbf{x}$ peuvent être intrinsèquement aléatoires. Leur distribution de probabilité doit être spécifiée à partir des éventuelles informations et données disponibles,
- **l'étape B'** est l'étape de V&V, éventuellement précédée d'une étape dite de *calage* du code dans le cas où le paramétrage $\boldsymbol{\theta}$ est incertain,
- **l'étape C** est l'étape de propagation des incertitudes à travers le code de calcul. Autrement dit, on cherche à quantifier l'incertitude affectant Q_y lorsque l'on fait varier les entrées du code.
- **l'étape C'** est l'étape d'analyse de sensibilité. On cherche à identifier les variables influentes du code en quantifiant l'impact de la variabilité de chacune des entrées sur Q_y.

L'approche prédictive (l'étape 1 de la procédure de validation comme nous l'avons définie précédemment) repose sur des techniques statistiques. Dans le paragraphe suivant, nous introduisons l'équation statistique générale reliant la réponse d'un code de calcul aux mesures du système physique d'intérêt. Nous la retrouverons déclinée sous différentes formes tout au long du mémoire.

Modélisation statistique

Les méthodes actuelles consacrées à l'approche prédictive font appel à des modèles statistiques reliant les expériences physiques recueillies avec les simulations. Ces méthodes sont dites non intrusives car elles n'ont pas besoin des équations mathématiques

qui sont à l'origine de l'élaboration du code. On parle aussi de méthodes *boîte-noire*. A l'heure actuelle, ces approches prédictives sont encore assez peu répandues. En effet, malgré leur caractère purement qualitatif, les comparaisons graphiques entre les mesures physiques et les simulations sont encore fréquemment utilisées par les ingénieurs et les physiciens car elles sont simples à mettre en œuvre, même si elles négligent les incertitudes. L'objectif de la modélisation statistique va donc être de quantifier l'incertitude de prédiction d'une réponse physique d'intérêt à l'aide des réponses d'un code de calcul. Bayarri et al. (2007) proposent dans cette direction une méthodologie destinée à la validation des codes de calculs qui consiste à calculer des intervalles de crédibilité autour de $r(\mathbf{x})$ en fonction de $\mathbf{x}$. La modélisation utilisée est très générale car elle suppose, d'une part que le vecteur de paramètres $\boldsymbol{\theta}$ du code est incertain et d'autre part qu'il n'existe pas de paramétrisation optimale, c'est à dire l'existence d'une valeur de $\boldsymbol{\theta}$ telle que $y_{\boldsymbol{\theta}}(\mathbf{x}) = r(\mathbf{x})$. L'absence de paramétrisation optimale nécessite l'introduction d'une fonction $b(\mathbf{x})$ appelée « erreur de code » telle que,

$$r(\mathbf{x}) = y_{\boldsymbol{\theta}}(\mathbf{x}) + b(\mathbf{x}) \tag{1.1}$$

où $\boldsymbol{\theta}$ est la valeur optimale mais inconnue du paramètre (dans un sens à définir) et qui doit donc être estimée. Nous verrons que l'interprétation du paramètre $\boldsymbol{\theta}$ peut être délicate du fait de phénomènes de compensation avec l'erreur $b(\mathbf{x})$ sur lesquels nous insisterons dans le chapitre 2. Les mesures physiques disponibles $z(\mathbf{x})$ (aussi appelées mesures expérimentales) sont, quant à elles, reliées au système physique via l'équation

$$z(\mathbf{x}) = r(\mathbf{x}) + \epsilon(\mathbf{x}), \tag{1.2}$$

où $\epsilon(\mathbf{x})$ est la variable aléatoire qui quantifie l'ensemble des erreurs qui dégradent la connaissance de $r(\mathbf{x})$, composée essentiellement des erreurs de mesures. En combinant les équations (1.1) et (1.2), nous obtenons l'équation statistique

$$z(\mathbf{x}) = y_{\boldsymbol{\theta}}(\mathbf{x}) + b(\mathbf{x}) + \epsilon(\mathbf{x}), \tag{1.3}$$

combinée avec une structure *a priori* sur l'erreur de code $b(\mathbf{x})$ qui tient compte de la plus grande régularité de cette fonction par rapport à la variable de bruit $\epsilon(\mathbf{x})$. Étant donné la complexité des systèmes étudiés, l'exécution du code peut ne pas être instantanée et nécessiter jusqu'à plusieurs heures, voire plusieurs jours de calculs. En pratique, cela se révèle être un obstacle très important pour l'estimation de $\boldsymbol{\theta}$ à partir de l'équation (1.3). Pour cette raison, une approximation du code est utilisée, appelée *émulateur* ou *métamodèle*, voire parfois *surface de réponse*. Elle fournit une prédiction de $y_{\boldsymbol{\theta}}(\mathbf{x})$ pour tout $\mathbf{x} \in \mathcal{X}$ à partir de seulement quelques réponses du code. Le processus gaussien est sûrement l'émulateur le plus utilisé en simulation numérique. Basé sur une représentation *a priori* de la réponse du code comme appartenant à la trajectoire d'un processus Gaussien, il a été introduit par Sacks et al. (1989). Un des objectifs de la thèse consistera en l'étude de la construction d'un tel émulateur pour le calage de code, qui consiste à réduire l'incertitude affectant $\boldsymbol{\theta}$.

Dans la littérature, le modèle (1.3) a démontré sa capacité à fournir de bonnes prédictions de $r(\mathbf{x})$. Mais, outre les difficultés relatives à l'inférence de $\boldsymbol{\theta}$ causées par les problèmes de non identifiabilité entre $\boldsymbol{\theta}$ et $b(\mathbf{x})$, les prédictions de $r(\mathbf{x})$ qui en découlent sont parfois difficiles à interpréter dans un contexte industriel. Un des objectif de la thèse consistera à apporter un éclairage précis sur l'utilisation du modèle (1.3) pour en faciliter l'utilisation par les analystes et les ingénieurs.

Une seconde classe de méthodes statistiques s'intéresse à la validation des codes de calculs $y_{\boldsymbol{\theta}}(\mathbf{x})$ dans un cadre davantage motivé par les contraintes industrielles. Les travaux

de Oberkampf et Barone (2006) sont inspirés des premières définitions du concept de V&V en mécanique des fluides. En particulier, ces auteurs proposent de quantifier l'écart entre les expériences physiques et les simulations en introduisant des métriques de validation basées sur des tests statistiques et satisfaisant des propriétés en accord avec les définitions que ces auteurs fournissent des concepts de validation et de calibration (Trucano et al., 2006). Ces méthodes seront présentées au chapitre 2. Nous mettrons en avant leurs avantages et leurs limites, qui nous conduisent à préférer des techniques de validation basées sur le modèle (1.3).

Contenu et organisation du mémoire

Le manuscrit compile plusieurs contributions dédiées au calage et à la validation des code de calculs coûteux (en temps de calcul) et pour lesquels les mesures physiques sont disponibles en nombre limité. Ces contributions se situent à plusieurs niveaux d'objectif, du calage vers l'étape 1 de la validation et jusqu'à l'utilisation effective du code (l'étape 2).

Dans le chapitre 2, nous dressons un état de l'art unifié des méthodologies existantes destinées au calage et à la validation des code de calculs.

Dans le chapitre 3, nous proposons d'appréhender l'étape 1 de la validation d'un code de calcul, d'abord comme un problème de sélection de modèle, plutôt que directement comme un problème de prédiction. Toujours dans un contexte d'inférence bayésienne, nous calculons le facteur de Bayes pour tester l'hypothèse

$$\mathrm{H}_0 : b(\mathbf{x}) = 0$$

contre l'hypothèse

$$\mathrm{H}_1 : b(\mathbf{x}) \neq 0.$$

Lorsque le code est linéaire en ses paramètres, c'est à dire lorsqu'il peut s'écrire sous la forme $y_{\boldsymbol{\theta}}(\mathbf{x}) = h(\mathbf{x})^{\mathrm{T}}\boldsymbol{\theta}$, des lois *a priori* choisies dans des classes partiellement conjuguées permettent, d'une part, de réduire la complexité des calculs, et d'autre part, de construire des procédures cohérentes de tests.

Dans le chapitre 4, nous proposons une contribution originale au calage d'un code de calcul coûteux dans un contexte d'inférence bayésienne. De nouveaux plans d'expériences séquentiels destinés à la construction d'un émulateur processus Gaussien sont présentés, basés sur le critère de l'amélioration espérée. Leur efficacité est illustrée sur des exemples académiques en plusieurs dimensions de l'espace des variables $\mathbf{x}$ et du paramètre $\boldsymbol{\theta}$.

Dans le chapitre 5, nous validons un code de thermique du bâtiment développé à EDF R&D. La méthode est mise en œuvre sur une séquence de 7 jours durant laquelle nous disposons de mesures physiques de puissance. Dans un deuxième temps, nous calculons quelle serait, du point de vue du fournisseur d'électricité, la prévision optimale de consommation énergétique moyenne du bâtiment au regard du type d'offre tarifaire souscrite par le client. Cette contribution à la garantie de performance énergétique illustre l'intérêt de la validation d'un code de calcul pour répondre à une problématique industrielle sous incertitude.

Bibliographie

AIAA (1998). Guide for the verification and validation of computational fluid dynamics simulations. *Reston VA : American Institute of Aeronautics and Astronautics*, AIAA-G-077-1998. 2

ASME (2009). *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*. American Society of Mechanical Engineers. 2

Bayarri, M. J., Berger, J. O., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., et Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2) :138–154. 2, 4

Oberkampf, W. et Barone, M. (2006). Measures of agreement between computation and experiment : Validation metrics. *Journal of Computational Physics*, 217(1) :5–36. 5

Roache, P. (1998). Verification of codes and calculations. *AIAA Journal*, 36(5) :696–702. 2

Sacks, J., W.J., W., Mitchell, T., et Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4) :409–423. 4

Santner, T., Williams, B., et Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag. 1

Trucano, T., Swiler, L., Igusa, T., Oberkampf, W., et Pilch, M. (2006). Calibration, validation and sensitivity analysis : What's what. *Reliability Engineering and System Safety*, 91(10-11) :1331–1357. 2, 5

Publications proches de la thèse

Damblin, G., Couplet, M., et Iooss, B. (2013). Numerical studies of space filling designs : optimization of latin hypercube samples and subprojection properties. *Journal of simulation*, 7 :276–289.

Damblin, G., Keller, M., Pasanisi, A., et Parent, E. (2014). Approche décisionnelle bayésienne pour estimer une courbe de fragilité. *Journal de la Société Française de Statistique*, 155(3).

Publications prévues

Damblin, G., Keller, M., Pasanisi, A., et Parent, E. (En cours de soumission). Bayesian model selection for the validation of computer codes.

Damblin, G., Keller, M., Pasanisi, A., et Parent, E. (Soumis). Adaptive numerical design for the calibration of computer codes.

Communications orales

Damblin, G., Barbillon, P., Keller, M., Pasanisi, A., et Parent, E. (2014). Plans d'expériences séquentiels pour la calibration des modèles numériques coûteux. 46èmes journées de Statistique, Rennes, 2-6 juin 2014.

Damblin, G., Barbillon, P., Keller, M., Pasanisi, A., et Parent, E. (2015a). Adaptive numerical designs for the calibration of computer models. MASCOT NUM 2015, 8-10 april.

Damblin, G., Barbillon, P., Keller, M., Pasanisi, A., et Parent, E. (2015b). Le facteur de Bayes appliqué à la validation des codes de calcul. 47èmes journées de Statistique, Lille, 1-5 juin 2015.

Communications posters

Damblin, G., Barbillon, P., Keller, M., Pasanisi, A., et Parent, E. (2014a). Adaptive numerical design for calibration of a computer model. UCM 2014, Uncertainty in Computer Model 2014 conference.

Damblin, G., Keller, M., Barbillon, P., Pasanisi, A., et Parent, E. (2014b). Bayesian validation of a computer model for the energy consumption of a building, with application to optimal bill pricing. Twelfth World Meeting of ISBA.

Damblin, G.and Keller, M., Barbillon, P., Pasanisi, A., et Parent, E. (2013). A Bayes decision approach to code validation in ana industrial context. MASCOT-SAMO 2013.

Chapitre 2

État de l'art

Sommaire

Ce chapitre d'état de l'art rend compte des problématiques majeures soulevées par la validation des codes de calcul dans un contexte industriel sous incertitudes. Dans la section 2.1, nous introduisons les notations utilisées dans ce mémoire pour désigner un système physique d'intérêt et sa modélisation à l'aide d'un code de calcul. Nous mettons en évidence l'ensemble des sources d'incertitudes pouvant affecter chacun de ces deux objets en distinguant l'incertitude induite par un aléa physique, appelée incertitude aléatoire, de l'incertitude traduisant un manque de connaissance, appelée incertitude épistémique.

Dans la section 2.2, nous commençons par définir le concept de validation d'un code de calcul en nous appuyant sur la littérature existante sur le sujet. Nous dressons un état de l'art succinct des méthodes de validation lorsque certaines variables d'entrée du code du code sont aléatoires. Ensuite, nous détaillons les méthodes fondées sur l'émulation par processus gaussien. Elles reposent sur une équation statistique reliant les réponses du code (aussi appelées simulations) avec les mesures du système physique d'intérêt.

Lorsque la valeur de certains paramètres du code est inconnue, la validation est souvent précédée d'une étape dite de « calage » du code qui consiste à mettre à jour leur incertitude à partir des données disponibles et dont font l'objet les sections 2.3 et 2.4. Nous mettons en évidence le besoin d'une procédure statistique pour tester la présence d'une fonction d'erreur entre la réponse du code et le système physique à prédire, appelée dans ce mémoire « erreur de code ». Une contribution à cette problématique fait l'objet du chapitre 3. Nous mettons aussi en évidence dans la section 2.3.2 l'imprécision du calage d'un code de calcul coûteux qui repose sur l'emulation par processus gaussien. La construction de plans d'expériences numériques spécifiques fait l'objet d'une contribution au chapitre 4. Nous terminons dans la section 2.5 par un aperçu des méthodes pour le calage et la validation des codes de calcul en grande dimension.

2.1 La simulation numérique d'un système physique

Dans les sections 2.1.1 et 2.1.2, nous introduisons les notations utilisées dans ce mémoire pour désigner le système physique d'intérêt et le code de calcul. Dans la section 2.1.3, nous identifions l'ensemble des sources d'incertitudes pouvant affecter à la fois les mesures physiques et les simulations, en distinguant l'incertitude aléatoire de l'incertitude épistémique.

2.1.1 Le système physique

Un système physique d'intérêt peut être assimilé dans sa forme la plus générale à une fonction réelle

$$r : \mathbf{x} = (\mathbf{x}^c, \mathbf{X}^e) \in \mathscr{X}^c \times \mathscr{X}^e \in \mathbb{R}^d \times \mathbb{R}^q \longrightarrow r(\mathbf{x}^c, \mathbf{X}^e) \in \mathbb{R}^m,$$

reliant un vecteur de variables expérimentales contrôlables $\mathbf{x}^c$ et un vecteur de variables incertaines $\mathbf{X}^e$, à une réponse physique d'intérêt $r(\mathbf{x}^c, \mathbf{X}^e)$ scalaire ou vectorielle [1]. Nous avons opté pour la notation en lettre capitale $\mathbf{X}^e$ pour mettre en évidence le caractère aléatoire de cette variable. Contrairement à $\mathbf{x}^c$ dont la valeur peut être imposée par le physicien chargé du dispositif expérimental, le vecteur $\mathbf{X}^e$ regroupe l'ensemble des variables que l'on ne peut pas contrôler *a priori*. Ces variables sont qualifiées de variables environnementales dans Santner et al. (2003). Pour illustrer la différence entre $\mathbf{x}^c$ et $\mathbf{X}^e$,

1. Le mémoire ne traite pas de la validation des codes de calcul à réponse fonctionnelle.

Santner et al. (2003) s'appuient sur l'exemple d'un code de calcul modélisant la charge induite par le port d'une prothèse de hanche sur les membres inférieurs. Le vecteur $\mathbf{x}^c$ regroupe les variables de conception du système, dont les dimensions de la prothèse, tandis que $\mathbf{X}^e$ regroupe l'amplitude et la direction de la force exercée sur la partie supérieure de la prothèse par la personne qui en est équipée, qui ne sont pas des quantités directement mesurables. Le système physique d'intérêt $r(\mathbf{x}^c, \mathbf{X}^e)$ est donc modélisé par une variable aléatoire dont nous noterons les réalisations $r(\mathbf{x}^c, \mathbf{x}^e)$ en gardant à l'esprit que la réalisation $\mathbf{x}^e \sim \mathbf{X}^e$ n'a pas été observée.

La mesure expérimentale, que l'on note $z(\mathbf{x}^c, \mathbf{x}^e)$ vérifie l'équation

$$z(\mathbf{x}^c, \mathbf{x}^e) = r(\mathbf{x}^c, \mathbf{x}^e) + \epsilon(\mathbf{x}^c, \mathbf{x}^e), \tag{2.1}$$

où $\epsilon(\mathbf{x}^c, \mathbf{x}^e)$ est une réalisation de la variable d'erreur $\epsilon(\mathbf{x}^c, \mathbf{X}^e)$, ce qui signifie que la mesure expérimentale $z(\mathbf{x}^c, \mathbf{x}^e)$ n'est pas une réalisation du système physique d'intérêt $r(\mathbf{x}^c, \mathbf{X}^e)$. La variable aléatoire $\epsilon(\mathbf{x}^c, \mathbf{X}^e)$ englobe l'ensemble des erreurs qui dégradent la connaissance de $r(\mathbf{x}^c, \mathbf{X}^e)$ et que nous détaillons dans la section 2.1.3. Introduisons préalablement les notations relatives au code de calcul qui tente de reproduire $r(\mathbf{x}^c, \mathbf{X}^e)$.

2.1.2 Le code de calcul

La simulation numérique complète l'expérience physique lorsque celle-ci est difficile à mettre en œuvre et/ou trop coûteuse, voire impossible à effectuer. Elle repose sur la modélisation du système physique d'intérêt sous la forme d'équations mathématiques traduisant la réalité physique. Par exemple, les codes de calcul de mécanique des fluides s'appuient sur les équations de *Navier-Stokes*, tandis que les codes de calcul simulant les phénomènes électromagnétiques s'appuient sur les équations de *Maxwell*. La solution de ces équations est implémentée dans un langage de programmation informatique menant à la création d'un code de calcul [2], représenté dans ce mémoire par la fonction paramétrée

$$y_{\boldsymbol{\theta}} : (\mathbf{x}^c, \mathbf{x}^e) \in \mathscr{X}^c \times \mathscr{X}^e \in \mathbb{R}^d \times \mathbb{R}^q \longrightarrow y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{x}^e) \in \mathbb{R}^m, \tag{2.2}$$

où $\boldsymbol{\theta} \in \mathscr{T} \subset \mathbb{R}^p$ est un vecteur de paramètres pouvant avoir une interprétation physique (voir chapitre 5). Notons que l'exécution du code nécessite d'attribuer une valeur à la variable environnementale $\mathbf{x}^e$ en entrée, bien qu'elle soit incertaine dans la réalité. Nous verrons dans la section 2.1.4.1 comment propager son incertitude à la réponse du code.

Nous nous intéressons dans ce mémoire à la validation des codes de calcul déterministes. Autrement dit, pour deux réalisations identiques des variables d'entrée et du paramètre $(\mathbf{x}_1^c, \mathbf{x}_1^e, \boldsymbol{\theta}_1) = (\mathbf{x}_2^c, \mathbf{x}_2^e, \boldsymbol{\theta}_2)$, alors $y_{\boldsymbol{\theta}_1}(\mathbf{x}_1^c, \mathbf{x}_1^e) = y_{\boldsymbol{\theta}_2}(\mathbf{x}_2^c, \mathbf{x}_2^e)$. De plus, nous utilisons le code comme une fonction boîte noire, ce qui signifie que les équations mathématiques reliant $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{x}^e)$ à $\mathbf{x}^c$ et $\mathbf{x}^e$ n'ont pas besoin d'être connues.

Enfin, nos contributions sont destinées à la validation des codes de calcul pour lesquels le paramètre $\boldsymbol{\theta}$ est incertain. Dans la section 2.1.3, nous détaillons les différentes sources d'incertitudes affectant à la fois le système physique et le code de calcul.

2.1.3 Les sources d'incertitudes

Pour valider un code de calcul, nous allons devoir quantifier sa capacité à prédire le système physique (voir section 2.2). Par conséquent, il est important de déterminer les

2. Dans la littérature, les appellations de modèle numérique ou simulateur sont également utilisées

sources d'incertitudes embarquées à la fois dans $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{x}^e)$ et dans la mesure expérimentale $z(\mathbf{x}^c, \mathbf{x}^e)$ (Kennedy et O'Hagan, 2001a). Nous verrons dans la section 2.2 que le traitement de l'incertitude dans les méthodes de validation des codes de calcul peut dépendre de sa nature, aléatoire ou épistémique.

Definition 2.1.3.1 *(**L'incertitude aléatoire**) L'incertitude aléatoire (aussi appelée incertitude intrinsèque, ou incertitude par essence) traduit une variabilité non-réductible.*

Definition 2.1.3.2 *(**L'incertitude épistémique**) L'incertitude épistémique désigne l'incertitude induite par un manque de connaissance (ou incertitude par ignorance). Contrairement à l'incertitude aléatoire, cette incertitude est réductible en présence d'informations supplémentaires.*

L'incertitude caractérisant les variables d'entrée $\mathbf{X}^e$ du système physique est aléatoire tandis que l'incertitude paramétrique induite par la méconnaissance de $\boldsymbol{\theta}$ est épistémique.

L'équation (2.1) indique que la variable aléatoire $z(\mathbf{x}^c, \mathbf{X}^e)$ n'est malheureusement pas égale au système physique $r(\mathbf{x}^c, \mathbf{X}^e)$. Ces deux quantités sont séparées par le terme d'erreur $\epsilon(\mathbf{x}^c, \mathbf{X}^e)$ qui regroupe deux sources d'incertitudes (Kennedy et O'Hagan, 2001a) :

1. l'erreur résiduelle qui provient de la non prise en compte de variables d'entrée dont l'influence sur $r(\mathbf{x}^c, \mathbf{X}^e)$ a été négligée. Elles sont parfois appelées variables de bruit,
2. les erreurs de mesure des variables contrôlées $\mathbf{x}^c$ et du système physique d'intérêt $r(\mathbf{x}^c, \mathbf{X}^e)$, qui dépendent du degré de précision de l'appareil de mesure.

L'erreur résiduelle est jugée épistémique car elle résulte d'une compréhension insuffisante du système. En effet, si certaines variables de bruit étaient identifiées, elles devraient faire partie du vecteur $\mathbf{X}^e$. L'erreur de mesure est par contre jugée aléatoire, car irréductible pour le scientifique qui relève la mesure. On pourrait cependant imaginer que la précision de l'appareil utilisé puisse être réduite par un technicien compétent, ce qui nous amènerait à considérer cette incertitude comme épistémique ! De même pour $\mathbf{X}^e$ dont on pourrait peut-être réduire l'incertitude grâce à une compréhension plus fine de la physique. Par conséquent, le statut aléatoire ou épistémique que l'on accorde à une source d'incertitude dépend des moyens alloués à l'étude du système physique et à sa modélisation (Kiureghian et Ditlevsen, 2009).

Remarque 2.1.3.1 *Lorsque la loi de probabilité attribuée à $\mathbf{X}^e$ est supposée paramétrique et doit être estimée (Barbillon et al., 2011), l'incertitude affectant les paramètres de la loi est épistémique.*

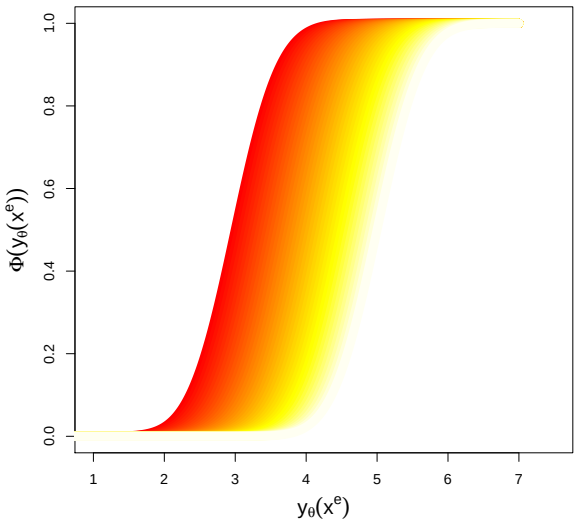
Dans la section suivante, nous allons détailler comment l'incertitude épistémique sur $\boldsymbol{\theta}$ et l'incertitude aléatoire $\mathbf{X}^e$ sont propagées à la réponse du code $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ et à la réponse physique $r(\mathbf{x}^c, \mathbf{X}^e)$. Cette étape de propagation d'incertitude constitue la première étape vers la validation du code (voir section 2.2).

2.1.4 Propagation des sources d'incertitudes

2.1.4.1 L'incertitude affectant la réponse du code de calcul

Pour propager l'incertitude aléatoire du vecteur $\mathbf{X}^e$ à la réponse du code de calcul, l'échantillonnage Monte-Carlo est la stratégie la plus naturelle à condition que l'on puisse disposer d'un nombre conséquent d'appels au code. La méthode consiste simplement à

FIGURE 2.1 – *Boîte de probabilité constituée des fonctions de répartition des lois gaussiennes de variance* 0.25 *lorsque la moyenne* $\boldsymbol{\theta}$ *varie dans l'intervalle* $[3,5]$.



simuler $\mathbf{x}_i^e$ dans sa distribution de probabilité, à exécuter le code en $\mathbf{x}_i^e$ et à récupérer les échantillons $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{x}_i^e)$ à partir desquels la distribution de probabilité de $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)$ peut être estimée.

Lorsque l'on veut propager ensemble, sans les confondre, l'incertitude épistémique sur $\boldsymbol{\theta}$ et l'incertitude du vecteur $\mathbf{X}^e$ à la réponse du code, Roy et Oberkampf (2011) proposent de représenter l'incertitude sur $\boldsymbol{\theta}$ par un ensemble de valeurs candidates et militent donc pour une représentation par intervalle de l'incertitude épistémique. Par ce biais, l'incertitude sur $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)$ est quantifiée par une boîte de probabilité, ou « p-box » dans la littérature anglo-saxonne (Ferson et al., 2003). La construction d'une boîte de probabilité repose sur la propagation de l'incertitude sur $\mathbf{X}^e$ à valeur de $\boldsymbol{\theta}$ fixée, ce processus étant répété pour un grand nombre de valeurs plausibles pour $\boldsymbol{\theta}$. Ainsi, l'incertitude globale affectant $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)$ peut être représentée par une famille de fonctions de répartition. À valeur de $\mathbf{x}^c$ fixée, la figure 2.1 illustre la boîte de probabilité obtenue en supposant que $y_\theta(\mathbf{X}^e) := y_\theta(\mathbf{x}^c,\mathbf{X}^e)$ suit une loi gaussienne de moyenne θ ayant pour plage d'incertitude l'intervalle $[3,5]$. La boîte de probabilité nous permet de distinguer la part d'incertitude provenant de l'aléa $\mathbf{X}^e$, caractérisée par sa forme, de la part d'incertitude provenant du manque de connaissance sur la valeur de $\boldsymbol{\theta}$, caractérisée par sa largeur. On pourrait penser, à tort, qu'un résultat similaire serait obtenu en spécifiant d'un point de vue bayésien une loi uniforme *a priori* sur $\boldsymbol{\theta}$, puis en propageant à travers le code l'incertitude affectant $\mathbf{X}^e$ conditionnellement à $\boldsymbol{\theta}$, le tout nécessitant une double boucle Monte-Carlo. L'article de Ferson et Ginzburg (1996) illustre les différences entre ces deux approches.

2.1.4.2 L'incertitude affectant la quantité physique d'intérêt

De la même façon que pour la réponse du code, l'incertitude affectant la réponse physique d'intérêt $r(\mathbf{x}^c,\mathbf{X}^e)$ peut être quantifiée à partir d'expériences physiques répétées en une même valeur du vecteur $\mathbf{x}^c$ $\{z(\mathbf{x}^c,\mathbf{x}_i^e)\}_{i=1}^r$ appelées répliques, à partir desquelles une distribution de probabilité de $r(\mathbf{x}^c,\mathbf{X}^e)$ peut être estimée. Malheureusement, le nombre d'expériences physiques possible est souvent limité, ce qui empêche bien souvent cette opération.

2.1.5 Systèmes physiques étudiés

Dans le chapitre 3, nous développons puis appliquons un nouvel outil de validation à un code de calcul modélisant les performances énergétiques d'une installation de production d'électricité. Dans le chapitre 5, nous étudions le comportement thermique d'un bâtiment prédit par un code de calcul dans le but de garantir des prévisions optimales de consommation moyenne du point de vue du fournisseur d'électricité et cela au regard d'un objectif de garantie de performance énergétique. La quantité d'intérêt $r(\mathbf{x}_t)(t)$ qui va nous intéresser est la puissance électrique injectée au temps t à l'intérieur d'un bâtiment en fonction du vecteur $\mathbf{x}_t$ des conditions contrôlées de l'expérience, comprenant notamment la température intérieure, la température extérieure, la vitesse du vent, les caractéristiques du bâtiment. Au département Enerbat d'EDF R&D, la plateforme expérimentale BESTLab permet la mesure de toutes ces quantités au cours du temps (Bontemps et al., 2013).

2.2 Vérification et Validation d'un code de calcul

Nous supposons que le domaine de variation des entrées du code $\mathscr{X}^c \times \mathscr{X}^e \subset \mathbb{R}^d \times \mathbb{R}^q$ constitue le domaine dans lequel on veut vérifier puis valider le code[3]. Bien que la vérification ne soit pas l'objet du travail rapporté dans ce mémoire, nous débutons cet état de l'art par la définition de ce concept. L'étape de validation, pour s'inscrire dans une procédure scientifique rigoureuse, doit être appliquée à des codes de calcul pour lesquels l'étape de vérification a été correctement effectuée.

2.2.1 Vérification d'un code de calcul

La vérification d'un code de calcul a pour objectif de quantifier l'erreur induite par la résolution approchée d'un système d'équations mathématiques pour lequel la solution exacte est inconnue (Roache, 1998). Elle est divisée en deux étapes distinctes. La première, appelée *vérification du code*, consiste à s'assurer que l'algorithme de résolution des équations a été correctement implémenté à l'aide d'un logiciel de programmation scientifique. En effet, plusieurs sources d'erreurs peuvent apparaître, que l'on peut imputer aux outils informatiques utilisés mais aussi aux procédures engagées pour construire le code. D'abord, la fiabilité du logiciel de programmation utilisé doit être examinée pour savoir si les données produites sont exactes. Ensuite, l'implémentation des algorithmes numériques de résolution sont potentiellement sources de « bugs », à identifier puis à corriger. La deuxième étape est appelée *vérification de la solution*. Elle se focalise principalement sur l'estimation de l'erreur de discrétisation qui correspond à l'écart entre la solution approchée du problème mathématique et la solution exacte, par exemple à partir de la méthode d'extrapolation de Richardson (Phillips et Roy, 2011).

La vérification du code et de la solution sont deux étapes indispensables pour pouvoir tirer des conclusions sur les résultats de l'étape de validation. En effet, si la validation détecte des écarts significatifs entre les mesures physiques et les simulations mais que le code n'a pas été vérifié, il est impossible d'établir si ces écarts proviennent d'une représentation mathématique insuffisante du système physique (ce que doit identifier l'étape de validation) ou si ces écarts proviennent d'une mauvaise vérification du code (voir l'exemple du code *ALEGRA* dans Trucano et al. (2006)).

3. Le terme domaine de validation est parfois utilisé.

2.2.2 Validation d'un code de calcul

La première définition de la validation d'un code de calcul a été donnée par la DMS0 (Defense Modeling and Simulation Office) aux États-Unis, puis reprise par le département ASC (Advanced Simulation and Computing program).

Definition 2.2.2.1 *La validation doit s'assurer que le code de calcul prédit correctement le système physique d'intérêt.*

Cette définition suggère une réponse par oui ou non après avoir visualisé graphiquement l'écart entre les simulations et les mesures physiques. Elle ne donne donc qu'un aperçu qualitatif de la capacité du code à reproduire le système physique sur $\mathscr{X}^c \times \mathscr{X}^e$. D'un point de vue opérationnel, il pourrait être plus judicieux de quantifier la proximité entre $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ et $r(\mathbf{x}^c, \mathbf{X}^e)$ en fonction de la valeur du vecteur $\mathbf{x}^c$. En effet, il est fort probable que pour certaines valeurs de $\mathbf{x}^c$, la réponse du code $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ soit très proche de $r(\mathbf{x}^c, \mathbf{X}^e)$, alors que pour d'autres valeurs ce ne soit pas le cas. Il se peut par exemple que dans le cadre d'une première étude de compréhension d'un système physique, une prédiction « grossière » du système physique soit suffisante (par exemple en moyenne sur $\mathscr{X}$), avant qu'une prédiction numérique plus précise ne soit requise pour certaines valeurs de $\mathbf{x}^c$. C'est pourquoi, une nouvelle définition tenant compte de l'utilisation du code de calcul a été proposée par la suite par l'AIAA (*American Institute of Aeronautics and Astronautics*) (AIAA, 1998) et reprise par l'ASME (*American Society of Mechanical Engineers*) (ASME, 2009).

Definition 2.2.2.2 *La validation doit quantifier avec quelle précision le code de calcul prédit le système physique d'intérêt et ceci en tenant compte de ses utilisations.*

Cette définition justifie l'approche en deux étapes que nous avons définie dans l'introduction du mémoire. La première étape fait appel à des méthodes statistiques fondées sur la comparaison des simulations $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ avec les mesures physiques disponibles $z(\mathbf{x}^c, \mathbf{X}^e)$. Son objectif est d'évaluer l'incertitude de prédiction du système physique $r(\mathbf{x}^c, \mathbf{X}^e)$ lorsque l'on utilise les réponses du code de calcul $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ et les mesures physiques. C'est la raison pour laquelle, elle est appelée *approche prédictive* (Bayarri et al., 2007b). La seconde étape consiste à décider si cette incertitude de prédiction est acceptable vis à vis de l'étude industrielle à mener. Dans la suite de cet état de l'art, le terme validation sera utilisé pour désigner la première étape puisque nous présentons les méthodes statistiques de quantification d'incertitudes du système physique indépendamment pour l'instant des enjeux industriels.

Nous allons maintenant introduire les deux approches statistiques les plus répandues dans la littérature pour valider un code de calcul. La première classe de méthodes (section 2.2.2.1) est issue de la littérature des sciences de l'ingénieur et s'appuie sur la construction d'une métrique de validation. La seconde classe de méthodes (section 2.2.2.2) s'appuie sur la modélisation par processus aléatoires gaussiens (Sacks et al., 1989). Dans les prochaines sections, nous présentons les principales caractéristiques de ces approches, leurs limites et similitudes. Nous situons également les contributions apportées dans ce manuscrit au regard de l'état de l'art effectué.

2.2.2.1 Les métriques de validation

Les métriques de validation permettent de quantifier l'écart entre les réponses du code $y_{\boldsymbol{\theta}}(\mathbf{x}^c, \mathbf{X}^e)$ et le système physique $r(\mathbf{x}^c, \mathbf{X}^e)$ (Oberkampf et Barone, 2006; Rebba, 2008;

Rebba et al., 2006). Rappelons la relation entre les mesures physiques et le système physique :

$$z(\mathbf{x}^c,\mathbf{x}^e) = r(\mathbf{x}^c,\mathbf{x}^e) + \epsilon(\mathbf{x}^c,\mathbf{x}^e).$$

La métrique de validation proposée dans Oberkampf et Barone (2006) s'écrit comme un test statistique qui permet d'encadrer à niveau de confiance fixé l'erreur

$$\mathbb{E}[y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)] - \mathbb{E}[r(\mathbf{x}^c,\mathbf{X}^e)] \tag{2.3}$$

à partir de répliques $\{z(\mathbf{x}^c,\mathbf{x}_i^e)\}_{i=1}^r$ qui constituent r réalisations de la variable aléatoire $z(\mathbf{x}^c,\mathbf{X}^e)$. Supposons maintenant que

$$\epsilon(\mathbf{x}^c,\mathbf{X}^e) \sim \mathcal{N}(0,\lambda^2),$$

et notons $\overline{z(\mathbf{x}^c,.)}$ la moyenne empirique du vecteur $z(\mathbf{x}^c,.) = (z(\mathbf{x}^c,\mathbf{x}_1^e),\cdots,z(\mathbf{x}^c,\mathbf{x}_r^e))$. Alors, la stratégie proposée consiste d'abord à utiliser la loi de la statistique de test de Student pour pouvoir écrire un encadrement de la quantité d'intérêt $\mathbb{E}[r(\mathbf{x},\mathbf{X}^e)]$:

$$\overline{z(\mathbf{x}^c,.)} - t_{m-1,1-\alpha/2} \times \frac{s}{\sqrt{r}} \leq \mathbb{E}[r(\mathbf{x},\mathbf{X}^e)] \leq \overline{z(\mathbf{x}^c,.)} + t_{m-1,1-\alpha/2} \times \frac{s}{\sqrt{r}} \tag{2.4}$$

où $t_{m-1,1-\alpha/2}$ est le quantile d'ordre $1-\alpha/2$ de la loi de Student à $m-1$ degrés de liberté et

$$s = \sqrt{\frac{1}{r-1}\sum_{i=1}^{r}\left(z(\mathbf{x}^c,\mathbf{x}_i^e) - \overline{z(\mathbf{x},.)}\right)^2}. \tag{2.5}$$

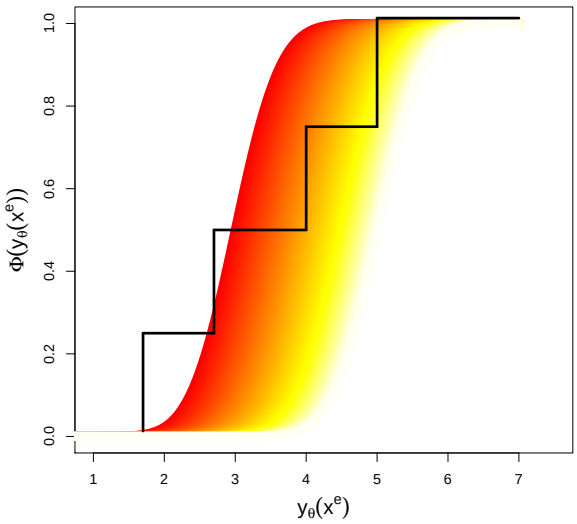
On obtient,

$$\begin{aligned}\mathbb{E}[y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)] - \overline{z(\mathbf{x}^c,.)} + t_{m-1,1-\alpha/2} \times \frac{s}{\sqrt{r}} &\geq \mathbb{E}[y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)] - \mathbb{E}[r(\mathbf{x}^c,\mathbf{X}^e)] \\ &\geq \mathbb{E}[y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)] - \overline{z(\mathbf{x}^c,.)} - t_{m-1,1-\alpha/2} \times \frac{s}{\sqrt{r}}\end{aligned} \tag{2.6}$$

ce qui fournit un encadrement de (2.3) à niveau de confiance $100(1-\alpha)\%$. Malheureusement, c'est la moyenne du système physique $\mathbb{E}[r(\mathbf{x},\mathbf{X}^e)]$ qui est comparée à la moyenne des réponses du code, alors que nous aimerions comparer les distributions de ces deux quantités.

C'est pourquoi, Roy et Oberkampf (2011) proposent une métrique de validation capable de comparer graphiquement des distributions de probabilité en tenant compte à la fois de l'incertitude aléatoire $\mathbf{X}^e$ et de l'incertitude épistémique affectant le paramètre $\boldsymbol{\theta}$. L'incertitude aléatoire portée par $\mathbf{X}^e$ est propagée par simulation Monte-Carlo tandis que l'incertitude affectant le paramètre $\boldsymbol{\theta}$ est exprimée par un intervalle de valeurs possibles. Ensuite, pour un certain nombre de valeurs $\boldsymbol{\theta_i}$ du paramètre $\boldsymbol{\theta}$ est calculée une variable aléatoire $y_{\boldsymbol{\theta_i}}(\mathbf{x}^c,\mathbf{X}^e)$ caractérisée par sa fonction de répartition conditionnellement à $\mathbf{x}^c$. La famille de fonctions de répartition ainsi obtenue est appelée boîte de probabilité ou p-box (voir section 2.1.4). À partir de répliques $\{z(\mathbf{x}^c,\mathbf{x}_i^e)\}_{i=1}^r$, la fonction de répartition empirique du système physique conditionnellement à $\mathbf{x}^c$ peut être établie puis comparée à la boîte de probabilité. Sur la figure 2.2, la boîte de probabilité ne recouvre pas entièrement la fonction de répartition empirique. Cette observation nous amène à penser que le code de calcul n'est pas capable de reproduire exactement le système physique. Autrement dit, aucune valeur de $\boldsymbol{\theta}$ ne peut donner l'égalité des distributions de probabilité $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)$ et $r(\mathbf{x}^c,\mathbf{X}^e)$. Cet écart est appelé erreur de code ou parfois biais de code, notion qui figure pour la première fois dans Kennedy et O'Hagan (2001a) (voir section 2.4.1).

FIGURE 2.2 – *Boîte de probabilité constituée des fonctions de répartition des lois gaussiennes de variance* 0.25 *lorsque la moyenne* $\boldsymbol{\theta}$ *varie dans l'intervalle* $[3,5]$. *En noir : fonction de répartition empirique des mesures physiques disponibles.*



Un premier inconvénient de ces deux méthodes de validation fondées sur des comparaisons directes entre les réponses du code et le système physique est qu'elles font appel à des répliques, rarement disponibles pour des systèmes physiques complexes. De plus, elles s'appliquent conditionnellement à une valeur de la variable contrôlée $\mathbf{x}^c$. Par conséquent, l'incertitude affectant

$$\mathbb{E}[y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)] - \mathbb{E}[r(\mathbf{x}^c,\mathbf{X}^e)]$$

ne nous donne aucune information sur l'incertitude affectant

$$\mathbb{E}[y_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^c,\mathbf{X}^e)] - \mathbb{E}[r(\tilde{\mathbf{x}}^c,\mathbf{X}^e)]$$

même lorsque $\mathbf{x}^c$ et $\tilde{\mathbf{x}}^c$ sont proches. Nous devons donc effectuer autant de tests statistiques qu'il y a de configurations d'intérêts du vecteur $\mathbf{x}^c$, ce qui n'est pas toujours possible lorsque les mesures physiques sont disponibles en nombre limité et qu'il n'est pas envisagé d'en produire de nouvelles.

Dans la prochaine section, nous dressons un état de l'art des approches par processus gaussien qui permettent d'estimer finement l'erreur de code pour des valeurs de $\mathbf{x}^c$ pour lesquelles aucune expérience physique n'a été effectuée, en tenant compte du fait que $y_{\boldsymbol{\theta}}(\mathbf{x}^c,\mathbf{X}^e)$ et $y_{\boldsymbol{\theta}}(\mathbf{x}^c+\delta\mathbf{x}^c,\mathbf{X}^e)$ peuvent être entachées d'une erreur similaire lorsque la réponse du code est suffisamment régulière. Nous avons étudié pour l'instant le cadre très général où certaines variables $\mathbf{X}^e$ du code étaient aléatoires. Toutefois, les approches par processus gaussien que nous présentons dans la section 2.2.2.2 sont mises en œuvre pour valider des codes de calcul pour lesquels l'incertitude aléatoire est nulle ou a été négligée : $\mathbf{X}^e = 0$. La validation revient alors à calculer l'incertitude de prédiction affectant la réponse physique $r(\mathbf{x}^c)$, désormais déterministe. Nous verrons que contrairement aux techniques fondées sur les métriques de validation, le calcul de l'incertitude de prédiction du système physique est fondé sur une équation statistique reliant les mesures physiques aux réponses du code.

2.2.2.2 Les approches par processus gaussien

Les méthodes exposées dans cette section supposent que $\mathbf{X}^e = 0$ mais aussi pour l'instant que le paramètre $\boldsymbol{\theta}$ est connu. Elles consistent à calculer des prédictions de $r(\mathbf{x})$ ($\mathbf{x} := \mathbf{x}^c$) pour tout $\mathbf{x} = (x^1, \cdots, x^d)^{\mathrm{T}} \in \mathscr{X}$ à partir des mesures physiques disponibles et des réponses du code de calcul. Supposons que l'on dispose d'un vecteur $\mathbf{z}^f$ constitué de n mesures expérimentales

$$\mathbf{z}^f := (z_1^f, \cdots, z_n^f)^{\mathrm{T}},$$

en fonction des conditions expérimentales

$$\mathbf{X}^f := \left[\mathbf{x}_1^f, \cdots, \mathbf{x}_n^f\right] \in \mathrm{M}_{d,n}(\mathbb{R}).$$

En supposant que l'erreur $\epsilon(\mathbf{x})$ se comporte comme un bruit blanc gaussien, l'équation (2.1) se réécrit pour tout $1 \leq i \leq n$,

$$z_i^f = r(\mathbf{x}_i^f) + \epsilon_i, \tag{2.7}$$

où

$$\epsilon_i \sim \mathscr{E}_i \underset{i.i.d.}{=} \mathscr{N}(0, \lambda^2) \tag{2.8}$$

sont des lois gaussiennes centrées, indépendantes et identiquement distribuées (i.i.d.), de variance λ^2, ce qui traduit le fait que l'amplitude de cette erreur est toujours la même indépendamment de la valeur de $\mathbf{x}$ et qu'il n'existe pas de lien entre les erreurs commises en deux valeurs différentes. Nous introduisons l'erreur de code $b(\mathbf{x})$ comme l'écart fonctionnel entre le système physique d'intérêt et la réponse du code :

$$\forall\, \mathbf{x} \in \mathscr{X},\ b(\mathbf{x}) = r(\mathbf{x}) - y(\mathbf{x}). \tag{2.9}$$

En combinant les équations (2.9) et (2.7), on obtient l'équation [4]

$$z_i^f = y(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i. \tag{2.10}$$

Posons

$$h_b(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_s(\mathbf{x}))^{\mathrm{T}},$$

où $h_1(.), \cdots, h_s(.)$ sont s fonctions de régressions. L'erreur de code $b(\mathbf{x})$ est considérée *a priori* comme la réalisation d'un processus aléatoire gaussien de moyenne $h_b(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}_b$, de variance σ_b^2 et de fonction de corrélation $\mathrm{K}_{\boldsymbol{\Psi}_b}(.,.)$:

$$b(\mathbf{X}^f) := \left(z(\mathbf{x}_1^f) - y(\mathbf{x}_1^f), \cdots, z(\mathbf{x}_n^f) - y(\mathbf{x}_n^f)\right)^{\mathrm{T}} \sim \mathscr{N}(h_b(\mathbf{X}^f)^{\mathrm{T}}\boldsymbol{\beta}_b, \sigma_b^2 \Sigma_{\boldsymbol{\Psi}_b}(\mathbf{X}^f) + \lambda^2 \mathbf{I}_n), \tag{2.11}$$

où

$$h_b(\mathbf{X}^f) = \left[h_b(\mathbf{x}_1^f), \cdots, h_b(\mathbf{x}_n^f)\right] \in \mathrm{M}_{p,n}(\mathbb{R}). \tag{2.12}$$

La moyenne $h_b(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}_b$ et la fonction de covariance $\sigma_b^2 \mathrm{K}_{\boldsymbol{\Psi}_b}(.,.)$ spécifient complètement la loi du processus. Le processus conditionné aux réalisations de l'erreur $b(\mathbf{X}^f)$, appelée processus *a posteriori* (Currin et al., 1991), est aussi un processus gaussien, noté

$$b(.)|b(\mathbf{X}^f) \sim \mathscr{PG}\left(\mu_{\boldsymbol{\beta}_b, \boldsymbol{\Psi}_b}(.), \mathrm{V}_{\sigma_b, \boldsymbol{\Psi}_b}(.,.)\right). \tag{2.13}$$

4. Kennedy et O'Hagan (2000) considèrent une formulation encore plus générale $z_i^f = \rho y(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i$ où ρ est un paramètre d'échelle entre le code et le système physique (voir section 2.4.1).

Les expressions de sa moyenne et de sa matrice de variance-covariance sont disponibles en annexe A.3. Il fournit une prévision probabiliste de $b(\mathbf{x})$ pour tout $\mathbf{x} \in \mathscr{X} \setminus \mathbf{X}^f$ et interpole les erreurs d'apprentissages $b(\mathbf{X}^f)$. En pratique, les hyperparamètres $\boldsymbol{\beta}_b, \sigma_b^2$ et $\boldsymbol{\Psi}_b$ sont rarement connus et doivent être estimés, par exemple en maximisant la vraisemblance gaussienne des erreurs $b(\mathbf{X}^f)$ (voir annexe A.3). Notons que la moyenne

$$\hat{b}(\mathbf{x}) = \mathbb{E}[b(\mathbf{x}) | b(\mathbf{X}^f)]$$

est un BLUP (*Best Linear Unbiaised Prediction*), c'est à dire un prédicteur sans biais de $b(\mathbf{x})$ et de variance minimale parmi tous les prédicteurs linéaires de $b(\mathbf{x})$ [5]. Autour de cette valeur moyenne, des intervalles d'incertitude sur $b(\mathbf{x})$ peuvent être calculés à partir de la variance du processus gaussien (2.13). Par suite, un prédicteur $\hat{r}(\mathbf{x})$ s'écrit

$$\forall\, \mathbf{x} \in \mathscr{X},\ \ \hat{r}(\mathbf{x}) = y(\mathbf{x}) + \hat{b}(\mathbf{x}), \tag{2.14}$$

et les intervalles d'incertitude autour de $\hat{r}(\mathbf{x})$ découlent directement des intervalles d'incertitude sur $\hat{b}(\mathbf{x})$.

Malheureusement, le calcul du prédicteur (2.14) pour tout $\mathbf{x} \in \mathscr{X}$ peut devenir infaisable lorsque le temps de calcul nécessaire pour l'exécution du code de calcul est prohibitif, typiquement si une simulation nécessite plusieurs heures, voire plusieurs jours. Dans ce cas, le code est dit « coûteux ». Pour surmonter cette contrainte, le code peut lui aussi être approché par un processus gaussien (Sacks et al., 1989). Dans ce contexte, il est appelé « émulateur » et s'écrit comme le processus gaussien conditionné par la connaissance d'un nombre M de réponses du code

$$y(\mathbf{D}_{\mathrm{M}}) = \left(y(\mathbf{x}_1^{\mathrm{D}}), \cdots, y(\mathbf{x}_{\mathrm{M}}^{\mathrm{D}})\right)^{\mathrm{T}}, \tag{2.15}$$

appelé base d'apprentissage, exécutées en fonction d'un nombre M de configurations du vecteur de variables d'entrée $\mathbf{x}$:

$$\mathbf{D}_{\mathrm{M}} = [\mathbf{x}_1^{\mathrm{D}}, \cdots, \mathbf{x}_{\mathrm{M}}^{\mathrm{D}}] \in \mathrm{M}_{d,\mathrm{M}}(\mathbb{R}). \tag{2.16}$$

$\mathbf{D}_{\mathrm{M}}$ est appelé plan d'expériences numériques (voir annexe A.4.1). Les fonctions de corrélations $\mathrm{K}_{\boldsymbol{\Psi}_y}(.,.)$ et $\mathrm{K}_{\boldsymbol{\Psi}_b}(.,.)$ sont choisies au regard des croyances *a priori* sur la régularité du code et de l'erreur de code, même si en pratique les fonctions gaussiennes et de Matern 5/2 sont les plus utilisées (voir annexe A.3). La description des procédures d'estimation des hyperparamètres des deux processus gaussiens $(\boldsymbol{\beta}_y, \sigma_y^2, \boldsymbol{\Psi}_y)$ et $(\boldsymbol{\beta}_b, \sigma_b^2, \boldsymbol{\Psi}_b)$, ainsi que l'expression analytique de $\hat{r}(\mathbf{x})$ sont disponibles dans Le Gratiet (2013), Wang et al. (2009).

Exemple d'application à une fonction analytique Nous proposons d'illustrer la méthode de validation par processus gaussien sur un exemple académique inspiré de Forrester et al. (2007) dans lequel la fonction

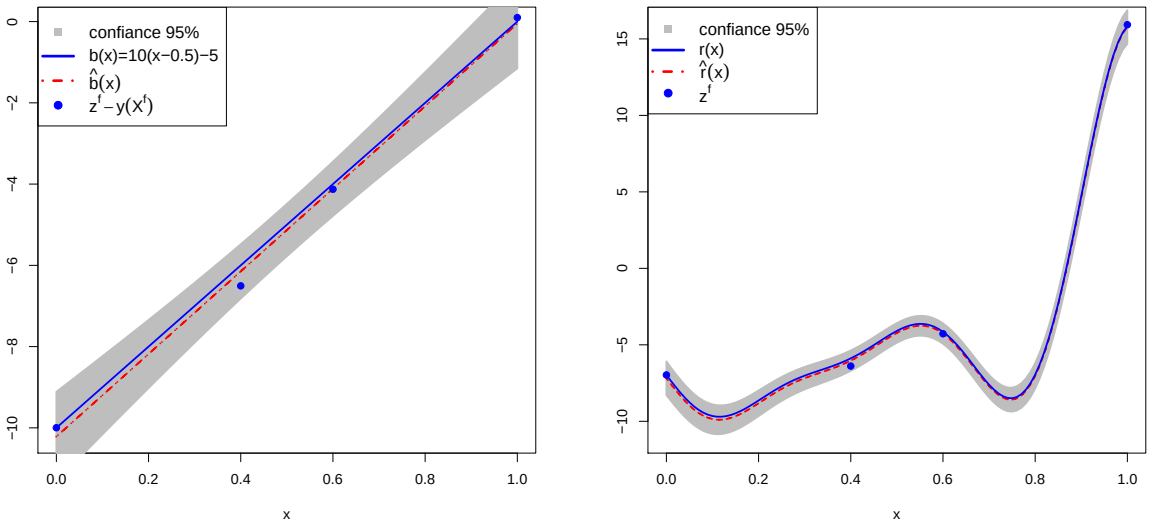
$$y(x) = (6x-2)^2 \sin(12x-4)$$

joue le rôle du code de calcul et la fonction

$$b(x) = 10(x-0.5) - 5$$

5. et même plus largement parmi tous les prédicteurs qu'ils soient linéaires ou non-linéaires, du fait de l'hypothèse gaussienne.

FIGURE 2.3 – *À gauche : prédiction par processus gaussien de $b(x)$ avec $h_b(x) = (1,x)^{\mathrm{T}}$ et une fonction de corrélation Matern 5/2. À droite : prédiction par processus gaussien de $r(x)$. Les hyperparamètres $\boldsymbol{\beta}_b, \sigma_b^2$ et Ψ_b sont estimés par maximum de vraisemblance en utilisant la librairie DiceKriging du logiciel R.*



joue le rôle de l'erreur de code. Supposons que $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$. Quatre mesures physiques $\mathbf{z}^f$ sont simulées suivant l'équation (2.10) où $\epsilon_i \sim \mathcal{N}(0, 0.3^2)$. Dans la procédure d'estimation, on suppose ici que $\lambda = 0.3$ est connu. La figure 2.3 illustre la prédiction de $b(x)$ (à gauche) et la prédiction de $r(x)$ (à droite). Sous l'hypothèse d'école que la fonction $y(x)$ est coûteuse, nous avons utilisé la librairie *MuFiCokriging* du logiciel R qui calcule le prédicteur proposé dans Le Gratiet (2013) avec M = 10 évaluations de la fonction $y(x)$ en réponse du plan d'expériences

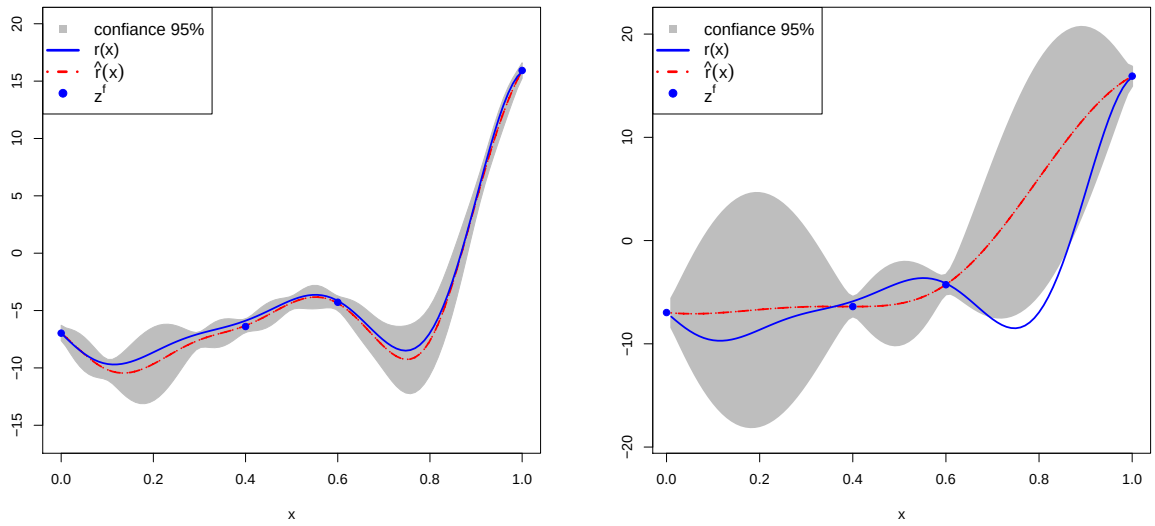
$$\mathbf{D}_{\mathrm{M}} = (0, 0.1, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1)^{\mathrm{T}}. \tag{2.17}$$

La figure 2.4 (à gauche) illustre la prédiction de $r(x)$ obtenue. Nous constatons que l'enveloppe de confiance est plus large du fait de l'incertitude supplémentaire induite par l'approximation de $y(x)$ par un processus gaussien, surtout pour les valeurs de la variable x où le code n'a pas été évalué (par exemple en $x = 0.2$).

La modélisation par processus gaussien de l'erreur et éventuellement de l'erreur et du code si les simulations sont coûteuses permet de répondre à la problématique de la validation. Elle permet de quantifier l'incertitude affectant la grandeur inconnue $r(x)$ à l'aide des mesures physiques disponibles et des réponses du code en considérant qu'il existe une erreur $b(x)$ entre le système physique et le code de calcul.

Extension à la multi-fidélité La figure 2.4 à droite illustre la prédiction de $r(\mathbf{x})$ fondée sur un modèle de régression des mesures physique $\mathbf{z}^f$ à l'aide d'un processus gaussien « bruité » (voir annexe A.3). On observe qu'elle est beaucoup moins précise que la prédiction obtenue avec la méthode décrite précédemment qui s'appuie à la fois sur les mesures physiques $\mathbf{z}^f$ qui renseignent directement sur $r(\mathbf{x})$ et sur les simulations $y(\mathbf{x})$ qui sont *a priori* moins précises mais plus faciles à obtenir (même si potentiellement coûteuses). On

FIGURE 2.4 – *À gauche : prédiction de $r(x)$ lorsque $y(x)$ et $b(x)$ sont modélisés par deux processus gaussiens avec $h_y(x) = 1$ et $h_b(x) = (1, x)^T$. Les structures de corrélation sont calculées à partir du noyau de matern 5/2. Les hyperparamètres des deux processus gaussiens $(\boldsymbol{\beta}_y, \sigma_y^2, \boldsymbol{\Psi}_y)$ et $(\boldsymbol{\beta}_b, \sigma_b^2, \boldsymbol{\Psi}_b)$ sont estimés par maximum de vraisemblance en utilisant la librairie MuFiCokriging du logiciel R. À droite : prédiction par processus gaussien de $r(x)$ à partir de $\mathbf{z}^f$ uniquement.*



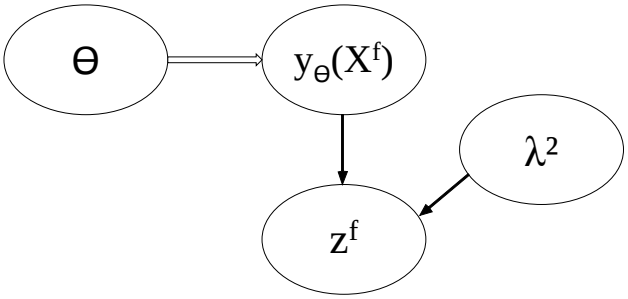
pourrait imaginer par extension de cette idée, disposer de versions de plus en plus simplifiées d'un code de calcul fournissant des prédictions de plus en plus grossières mais de moins en moins coûteuses et les utiliser conjointement pour améliorer la prédiction de $r(\mathbf{x})$. On parle alors de modélisation multi-fidélité. Si l'on disposait de N codes de calcul, de la version la plus dégradée à la version la plus précise, alors pour $1 \leq j \leq N$ on pourrait considérer le système d'équations

$$\begin{cases} z_i^f = y^N(\mathbf{x}_i^f) + b^N(\mathbf{x}_i^f) + \epsilon_i \\ y^j(\mathbf{x}_i^f) = \rho^j(\mathbf{x}_i^f)\, y^{j-1}(\mathbf{x}_i^f) + b^{j-1}(\mathbf{x}_i^f) \end{cases} \tag{2.18}$$

Les paramètres $\rho^j(\mathbf{x})$ sont introduits pour capturer d'éventuels changements d'échelles plus ou moins complexes entre deux niveaux de code. Qian et Wu (2008) les modélise encore par un processus gaussien tandis que Kennedy et O'Hagan (2000) et Le Gratiet (2013) les modélisent par une constante. Il se peut qu'une assez bonne connaissance des codes par les ingénieurs et numériciens chargés de les développer puisse permettre de choisir *a priori* la forme des paramètres $\rho^j(\mathbf{x})$.

Dans la suite de ce chapitre d'état de l'art, nous nous intéressons à la validation des codes de calcul $y_{\boldsymbol{\theta}}(\mathbf{x})$ lorsque le paramètre $\boldsymbol{\theta}$ est incertain. L'estimation de $\boldsymbol{\theta}$, appelée *calage* du code, pourra être effectuée à partir des mesures physiques disponibles $\mathbf{z}^f$ et des simulations conjointement à la validation. Dans la section 2.3, nous supposons que $b(\mathbf{x}) = 0$. Dans la section 2.4, nous étendrons la notion de calage d'un code de calcul lorsque l'on tient compte de l'erreur de code entre le système physique $r(\mathbf{x})$ et la réponse du code de calcul $y_{\boldsymbol{\theta}}(\mathbf{x})$.

FIGURE 2.5 – *Réseau bayésien (DAG) correspondant au modèle statistique (2.20).*



2.3 Calage des codes de calcul

Le calage statistique d'un code de calcul consiste à mettre à jour l'incertitude affectant la valeur du paramètre incertain $\boldsymbol{\theta}$ à partir des mesures disponibles de la quantité physique d'intérêt $\mathbf{z}^f$ (Campbell, 2006). Comme dans la section précédente, le calage d'un code de calcul est conditionné par le choix d'une équation statistique reliant la réponse du code $y_{\boldsymbol{\theta}}(\mathbf{x})$ aux mesures $\mathbf{z}^f$. Lorsque l'erreur de code $b(\mathbf{x})$ est négligée, le calage consiste à estimer $\boldsymbol{\theta}$ de telle sorte que $y_{\boldsymbol{\theta}}(\mathbf{x})$ reproduise aussi précisément que possible le système physique d'intérêt. On parle alors de calage sans erreur (ou sans biais) (Cox et al., 2001). Lorsque le calage est effectué en présence d'une erreur structurelle $b(\mathbf{x})$, il consiste à estimer conjointement le couple $(\boldsymbol{\theta}, b(\mathbf{x}))$. Dans cette section, nous illustrons les caractéristiques du calage de code sans erreur.

2.3.1 Modélisation statistique

Nous commençons par illustrer le calage des paramètres d'un code de calcul lorsque le code est capable de prédire parfaitement le système physique (Cox et al., 2001). Cette hypothèse se traduit mathématiquement par l'équation suivante :

$$\exists \boldsymbol{\theta} \in \mathscr{T} \ ; \ \forall \mathbf{x} \in \mathscr{X} \quad r(\mathbf{x}) = y_{\boldsymbol{\theta}}(\mathbf{x}) \tag{2.19}$$

En combinant l'équation (2.19) avec l'équation (2.7), on obtient pour $1 \leq i \leq n$,

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + \epsilon_i \tag{2.20}$$

L'équation (2.20) est un modèle de régression statistique des mesures physiques sur la réponse du code. Dans un contexte d'estimation bayésienne, la formule de Bayes combine l'information *a priori* $\pi(\boldsymbol{\theta}, \lambda^2)$ sur le couple de paramètres $(\boldsymbol{\theta}, \lambda^2)$ avec l'information apportée par $\mathbf{z}^f$ au travers de la fonction de vraisemblance $\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta}, \lambda^2)$. Le produit de ces deux quantités est proportionnelle à la distribution *a posteriori*,

$$\pi(\boldsymbol{\theta}, \lambda^2|\mathbf{z}^f) \propto \mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta}, \lambda^2)\pi(\boldsymbol{\theta}, \lambda^2). \tag{2.21}$$

La figure 2.5 représente le DAG (*Directed Acyclic Graph*) [6] correspondant au modèle statistique (2.20), ayant pour vraisemblance

$$\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta}, \lambda^2) = \frac{1}{(\sqrt{2\pi}\lambda)^n} \exp\left[-\frac{1}{2\lambda^2}\mathrm{SS}(\boldsymbol{\theta})\right] \tag{2.22}$$

6. voir par exemple l'ouvrage de Denis et Scutari (2014).

où

$$\mathrm{SS}(\boldsymbol{\theta}) = ||\mathbf{z}^f - y_{\boldsymbol{\theta}}(\mathbf{X}^f)||^2.$$

est le carré de la norme euclidienne du vecteur des écarts entre les mesures physiques et les réponses du code, aussi appelé vecteur des résidus. Soit

$$h(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_p(\mathbf{x}))^{\mathrm{T}}$$

un vecteur de fonctions de régressions évalué en $\mathbf{x} \in \mathscr{X}$. Lorsque le code est linéaire en $\boldsymbol{\theta}$, c'est à dire lorsque

$$y_{\boldsymbol{\theta}}(\mathbf{X}^f) = h(\mathbf{X}^f)^{\mathrm{T}}\boldsymbol{\theta} \tag{2.23}$$

où

$$h(\mathbf{X}^f) = \left[h(\mathbf{x}_1^f), \cdots, h(\mathbf{x}_n^f)\right] \in \mathrm{M}_{p,n}(\mathbb{R}) \tag{2.24}$$

la loi *a posteriori* (2.21) est analytique pour certaines lois *a priori* $\pi(\boldsymbol{\theta}, \lambda^2)$ bien choisies (la loi conjuguée gamma-normale ou la loi non informative de Jeffrey). Cependant, en pratique, les codes de calcul modélisent bien souvent des phénomènes physiques complexes aux comportements non-linéaires. Dans ce cas, la loi *a posteriori* n'admet pas d'expression explicite. Cette première difficulté s'associe souvent à un coût élevé des simulations. Dans ce cas, la vraisemblance (2.22) devient coûteuse à évaluer ne permettant pas d'envisager l'échantillonage de (2.21) avec des méthodes MCMC [7] (Robert, 1996). Pour contourner cet obstacle, le code peut être modélisé *a priori* par un émulateur processus gaussien de moyenne $h_y(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}_y$, de variance σ_y^2 et de fonction de corrélation $\mathrm{K}_{\boldsymbol{\Psi}_y}(.)$ (voir annexe A.3). Il est estimé à partir d'un nombre restreint M de simulations du code

$$y(\mathbf{D}_{\mathrm{M}}) = \left(y_{\boldsymbol{\tau}_1^{\mathrm{D}}}(\mathbf{x}_1^{\mathrm{D}}), \cdots, y_{\boldsymbol{\tau}_{\mathrm{M}}^{\mathrm{D}}}(\mathbf{x}_{\mathrm{M}}^{\mathrm{D}})\right)^{\mathrm{T}}$$

exécutées en un plan d'expériences de $\mathscr{X} \times \mathscr{T}$

$$\mathbf{D}_{\mathrm{M}} = \left[(\mathbf{x}_1^{\mathrm{D}}, \boldsymbol{\tau}_1^{\mathrm{D}}), \cdots, (\mathbf{x}_{\mathrm{M}}^{\mathrm{D}}, \boldsymbol{\tau}_{\mathrm{M}}^{\mathrm{D}})\right] \in \mathrm{M}_{d+p,\mathrm{M}}(\mathbb{R}).$$

Et donc,

$$y(\mathbf{D}_{\mathrm{M}}) \sim \mathscr{N}(h_y(\mathbf{D}_{\mathrm{M}})^{\mathrm{T}}\boldsymbol{\beta}_y, \sigma_y^2 \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}})) \tag{2.25}$$

où $\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}}) := \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}}, \mathbf{D}_{\mathrm{M}})$ et σ_y^2 est la variance du processus. D'après l'équation (2.20), les mesures physiques $\mathbf{z}^f$ contribuent à l'estimation du processus gaussien. Il nous faut donc considérer la loi jointe des simulations et des mesures physique $\mathbf{d} = (y(\mathbf{D}_{\mathrm{M}}), \mathbf{z}^f)$ pour estimer les hyperparamètres $\boldsymbol{\beta}_y$, σ_y^2 et $\boldsymbol{\Psi}_y$ du processus gaussien et le paramètre $\boldsymbol{\theta}$. En faisant apparaître les réponses latentes (non observées) du code $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$, la vraisemblance complète s'écrit

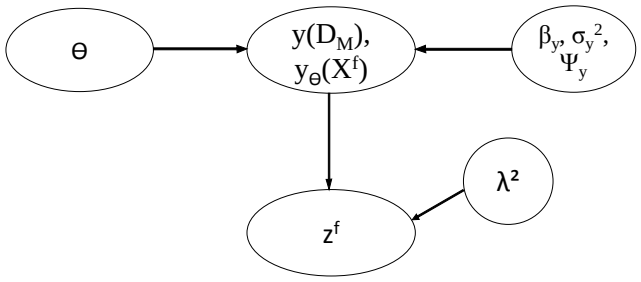
$$\begin{aligned}\mathscr{L}(\mathbf{d}, y_{\boldsymbol{\theta}}(\mathbf{X}^f)|\boldsymbol{\theta}, \lambda^2, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) = \mathscr{L}(\mathbf{z}^f|\lambda^2, y_{\boldsymbol{\theta}}(\mathbf{X}^f), \boldsymbol{\theta}) \\ \mathscr{L}(y_{\boldsymbol{\theta}}(\mathbf{X}^f)|y(\mathbf{D}_{\mathrm{M}}), \boldsymbol{\theta}, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y)\mathscr{L}(y(\mathbf{D}_{\mathrm{M}})|\sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y).\end{aligned} \tag{2.26}$$

La figure 2.6 représente le DAG correspondant à (2.26). La loi marginale de $\mathbf{d}$ est calculée en intégrant par rapport à la loi du vecteur $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$:

$$\begin{aligned}\mathscr{L}(\mathbf{d}|\boldsymbol{\theta}, \lambda^2, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) = \int \mathscr{L}(\mathbf{z}^f|\lambda^2, y_{\boldsymbol{\theta}}(\mathbf{X}^f), \boldsymbol{\theta})\mathscr{L}(y_{\boldsymbol{\theta}}(\mathbf{X}^f)|y(\mathbf{D}_{\mathrm{M}}), \boldsymbol{\theta}, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) \\ \mathscr{L}(y(\mathbf{D}_{\mathrm{M}})|\sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y)\,\mathrm{d}y_{\boldsymbol{\theta}}(\mathbf{X}^f),\end{aligned} \tag{2.27}$$

7. *Markoc Chain Monte Carlo qui nécessitent des milliers d'évaluation de la vraisemblance (Méthodes de Monte Carlo par chaînes de Markov)*

FIGURE 2.6 – *Réseau bayésien (DAG) correspondant au modèle (2.20) où le code de calcul est remplacé par un processus gaussien d'hyperparamètres* $(\boldsymbol{\beta}_y, \sigma_y^2, \boldsymbol{\Psi}_y)$.



avec

$$\mathbf{z}^f|\lambda^2, y_{\boldsymbol{\theta}}(\mathbf{X}^f), \boldsymbol{\theta} \sim \mathcal{N}(y_{\boldsymbol{\theta}}(\mathbf{X}^f), \lambda^2 \mathrm{I}_n), \tag{2.28}$$

et

$$y_{\boldsymbol{\theta}}(\mathbf{X}^f)|y(\mathbf{D}_\mathrm{M}), \boldsymbol{\theta}, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y \sim \mathcal{N}\left(\mu_{\boldsymbol{\beta}_y, \boldsymbol{\Psi}_y}(\mathbf{D}_{\boldsymbol{\theta}}), \mathrm{V}_{\sigma_y^2, \boldsymbol{\Psi}_y}(\mathbf{D}_{\boldsymbol{\theta}})\right) \tag{2.29}$$

où $\mathbf{D}_{\boldsymbol{\theta}} = \left[(\mathbf{x}_1^f, \boldsymbol{\theta}), \cdots, (\mathbf{x}_n^f, \boldsymbol{\theta})\right] \in \mathrm{M}_{d+p,n}(\mathbb{R})$ et les expressions de la moyenne et de la matrice de variance-covariance de (2.29) sont disponibles en annexe A.3. L'intégrale (2.27) est calculée par rapport à la mesure de Lesbegue sur $\mathbb{R}$. On obtient,

$$\mathscr{L}(\mathbf{d}|\boldsymbol{\theta}, \lambda^2, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) = \mathscr{L}(\mathbf{z}^f|\lambda^2, y(\mathbf{D}_\mathrm{M}), \boldsymbol{\theta}, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) \times \mathscr{L}(y(\mathbf{D}_\mathrm{M})|\sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y). \tag{2.30}$$

En utilisant d'une part, le fait que le produit de deux densités gaussiennes est une densité gaussienne à une constante de normalisation près, et d'autre part que cette constante de normalisation est également une densité gaussienne (voir annexe A.2), on peut montrer que

$$\mathscr{L}(\mathbf{d}|\boldsymbol{\theta}, \lambda^2, \sigma_y^2, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y) \propto \frac{|\mathrm{C}_{\boldsymbol{\Psi}_y}|^{-1/2}}{\sigma_y^{n+\mathrm{M}}} \exp -\frac{1}{2\sigma_y^2}\Big[\left(\mathbf{d} - (h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_y, h(\mathbf{D}_{\boldsymbol{\theta}})^\mathrm{T}\boldsymbol{\beta}_y)\right)^\mathrm{T} \mathrm{C}_{\boldsymbol{\Psi}_y}^{-1}\left(\mathbf{d} - (h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_y, h(\mathbf{D}_{\boldsymbol{\theta}})^\mathrm{T}\boldsymbol{\beta}_y)\right)\Big],$$

avec

$$\mathrm{C}_{\boldsymbol{\Psi}_y} = \begin{pmatrix} \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_\mathrm{M}) & \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_\mathrm{M}, \mathbf{D}_{\boldsymbol{\theta}}) \\ \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_\mathrm{M}, \mathbf{D}_{\boldsymbol{\theta}}) & \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\boldsymbol{\theta}}) + \frac{\lambda^2}{\sigma_y^2}\mathbf{I}_n \end{pmatrix}.$$

Alors que le choix de la fonction de covariance peut s'effectuer en utilisant les connaissances *a priori* éventuelles sur la régularité de la réponse du code, les hyperparamètres σ_y^2, $\boldsymbol{\beta}_y$ et $\boldsymbol{\Psi}_y$ sont très souvent inconnus et doivent être estimés à partir de $\mathbf{d}$. En pratique, il est préférable de séparer l'estimation de $\boldsymbol{\theta}$ de l'estimation des hyperparamètres. Suivant l'approche suggérée par Cox et al. (2001), les hyperparamètres sont d'abord estimés en maximisant la vraisemblance partielle des simulations $\mathscr{L}^\mathrm{M}$, donnée par

$$\mathscr{L}^\mathrm{M}(y(\mathbf{D}_\mathrm{M})|\sigma_y, \boldsymbol{\beta}_y, \boldsymbol{\Psi}_y, \mathbf{D}_\mathrm{M}) \propto \frac{|\Sigma_{\boldsymbol{\Psi}_y}|^{-1/2}}{\sigma_y^\mathrm{M}} \exp -\frac{1}{2\sigma_y^2}\Big[(y(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_y)^\mathrm{T} \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_\mathrm{M})^{-1}(y(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_y)\Big], \tag{2.31}$$

puis l'estimation de $\boldsymbol{\theta}$ s'effectue à partir de la loi *a posteriori*

$$\pi^{\mathrm{C}}(\boldsymbol{\theta}, \lambda^2 | \mathbf{d}) \propto \mathscr{L}^{\mathrm{C}}(\mathbf{z}^f | \boldsymbol{\theta}, \lambda^2, y(\mathbf{D}_{\mathrm{M}})) \pi(\boldsymbol{\theta}, \lambda^2), \tag{2.32}$$

basée sur la vraisemblance des mesures physiques $\mathbf{z}^f$ conditionnellement à $y(\mathbf{D}_{\mathrm{M}})$

$$\mathscr{L}^{\mathrm{C}}(\mathbf{z}^f | \boldsymbol{\theta}, y(\mathbf{D}_{\mathrm{M}}), \lambda^2) \propto |\mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{D}_{\boldsymbol{\theta}}) + \lambda^2 \mathbf{I}|^{-1/2} \exp -\frac{1}{2} \Big[\big(\mathbf{z}^f - \boldsymbol{\mu}^{\mathbf{M}}_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{D}_{\boldsymbol{\theta}})\big)^{\mathrm{T}} \\ (\mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{D}_{\boldsymbol{\theta}}) + \lambda^2 \mathbf{I}_n)^{-1} \big(\mathbf{z}^f - \boldsymbol{\mu}^{\mathbf{M}}_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{D}_{\boldsymbol{\theta}})\big) \Big]$$

où $(\hat{\boldsymbol{\beta}}_y, \hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y)$ est un estimateur du maximum de la vraisemblance (2.31). Cette procédure en deux étapes qui néglige l'impact de $\mathbf{z}^f$ sur l'estimation de $\boldsymbol{\theta}$ est appelée « approche bayésienne modulaire » (Liu et al., 2009). La loi *a posteriori* (2.32) peut être évaluée instantanément, rendant possible son échantillonnage à l'aide d'algorithmes MCMC. Dans le chapitre 4, nous montrons sous certaines hypothèses peu restrictives la convergence uniforme sur $\mathscr{T}$ de la loi *a posteriori* approchée (2.32) vers la loi *a posteriori* cible (2.21) quand $\mathrm{M} \to \infty$ lorsque $\mathscr{T}$ est borné. Par conséquent, plus l'on dispose d'un nombre important de simulations et plus la loi a *posteriori* approchée est proche de la loi a *posteriori* cible (2.21).

Dans l'exemple académique qui suit, nous comparons l'échantillonnage de (2.21) avec l'échantillonnage de (2.32). Nous nous intéressons à l'impact de la taille du plan d'expériences numériques $\mathbf{D}_{\mathrm{M}}$ sur la proximité entre ces deux lois.

2.3.2 Impact du plan d'expériences numériques

Pour illustrer l'importance du plan d'expériences sur l'erreur commise lorsque l'on échantillonne la loi *a posteriori* approchée (2.32) à la place de la loi a *posteriori* cible (2.21), supposons que la fonction analytique

$$y_\tau : x \in [0,1] \longrightarrow (6x-2)^2 \times \sin(\tau x - 4) \tag{2.33}$$

joue de nouveau le rôle du code de calcul. Soit $\mathbf{X}^f = (0.1, 0.3, 0.8)^{\mathrm{T}}$. Nous simulons des mesures physiques $\mathbf{z}^f$ suivant l'équation (2.20) avec $\theta = 12$ et $\lambda = 0.3$. Dans la procédure d'estimation décrite dans la section précédente, nous supposons toujours que λ est connu (fixé ici à 0.3) si bien que seul le paramètre θ est à estimer. Dans un premier temps, nous calculons (2.21) à l'aide d'un algorithme de Métropolis-Hastings en choisissant la loi *a priori* uniforme

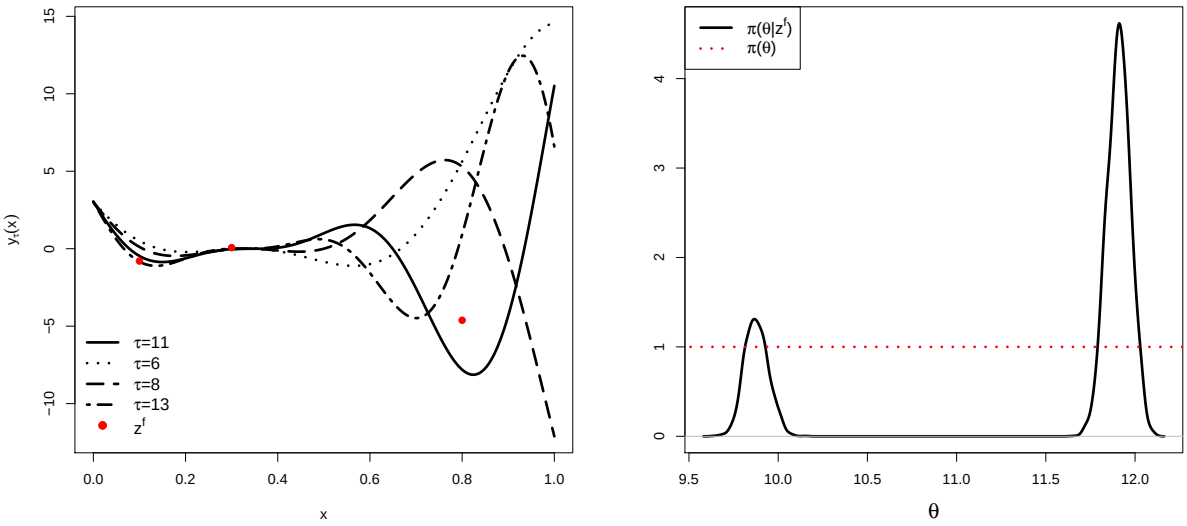
$$\pi(\theta) \propto 1_{[5,15]}.$$

La figure 2.7 (à droite) montre une densité de la loi *a posteriori* cible à deux modes, l'un autour de la vraie valeur $\theta \approx 12$ à partir de laquelle ont été simulées les mesures $\mathbf{z}^f$ et l'autre autour de $\theta \approx 9.8$.

Maintenant, on échantillonne la loi *a posteriori* approchée (2.32) pour plusieurs tailles de plan d'expériences LHD maximin[8]. La figure 2.8 illustre les densités de probabilités obtenues. On constate que pour un nombre restreint de simulations $\mathrm{M} = 45$ et $\mathrm{M} = 60$, elles sont très éloignées de la loi cible. Lorsque M augmente, dans notre exemple lorsque $\mathrm{M} = 75$ et $\mathrm{M} = 90$, alors les régions autour de $\theta = 12$ et $\theta = 9.8$ sont clairement identifiées. Ces résultats illustrent que plus la taille du plan d'expériences est élevée, plus la distribution *a posteriori* (2.32) ressemblera à (2.21). Toutefois, même à taille de plan fixée, on

8. Latin Hypercube Design maximin

FIGURE 2.7 – *À gauche : la fonction jouet* $y_\tau(x) = (6x-2)^2 \times \sin(\tau x - 4)$ *pour plusieurs valeurs de* τ. *À droite : la densité de probabilité de la distribution a posteriori (2.21).*



s'aperçoit que la loi *a posteriori* approchée est fortement dépendante de l'émulateur et par conséquent du plan d'expériences utilisé. La figure 2.9 illustre 4 calages basés sur π^C à partir d'un émulateur à chaque fois différent provenant d'un plan d'expériences LHD maximin de taille M = 50 à chaque fois différent (car construit de façon aléatoire à partir d'un algorithme de recuit simulé [9]). Nous pouvons observer que les 4 densités de probabilité π^C sont très différentes les unes des autres et pour certaines très éloignées de la loi cible π. Pourtant, la qualité prédictive de chacun des émulateurs à l'origine de chacune de ces lois est satisfaisante car leur coefficient de prédictivité R^2 est supérieur à 0.80 (voir annexe A.3 pour la définition du R^2). Cela illustre que les plans d'expériences qui remplissent de façon optimale le domaine $\mathscr{X} \times \mathscr{T}$ comme les LHD maximin ne sont pas les plans les mieux adaptés pour le calage de code où l'on a besoin, pour que π^C soit proche de π, de construire un émulateur processus gaussien le plus précis possible mais seulement pour les valeurs du paramètre $\boldsymbol{\theta}$ où la loi *a posteriori* cible $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$ est élevée.

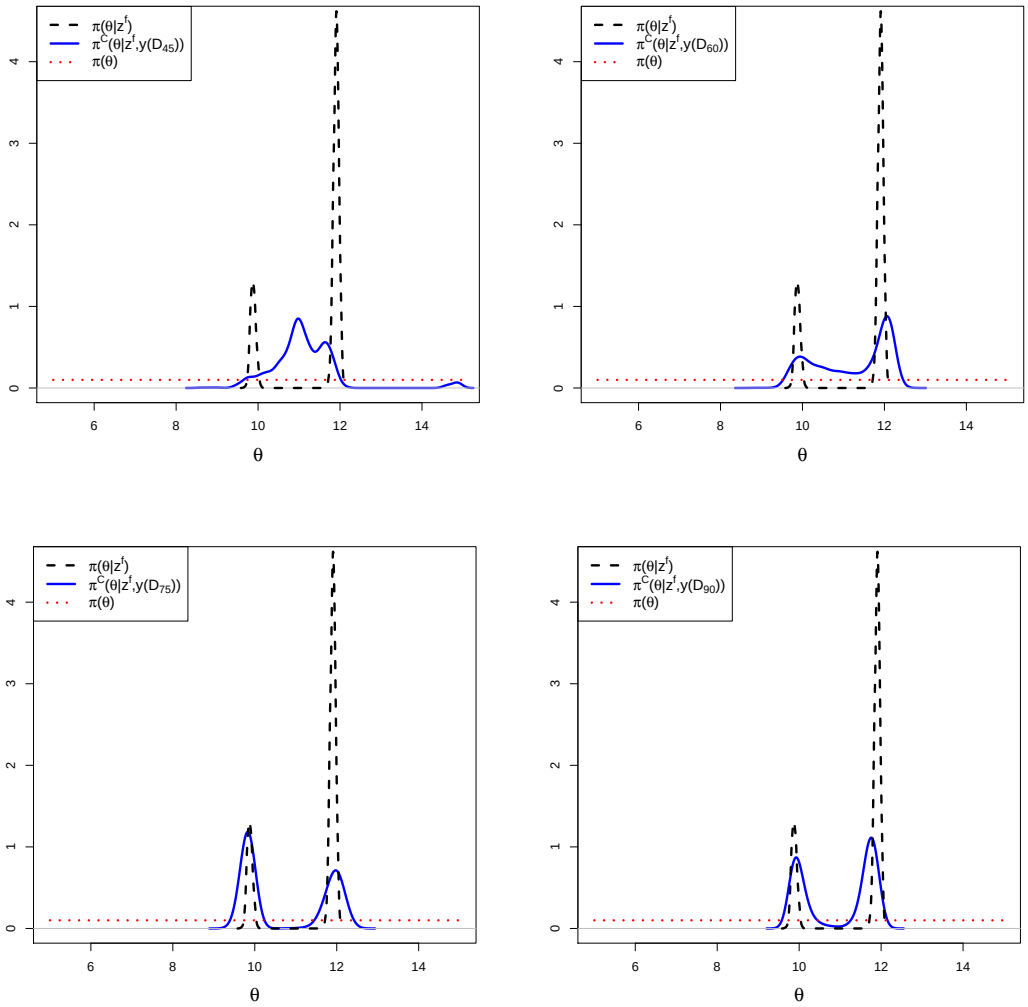
Même si graphiquement sur nos simulations, on peut se rendre compte du large panel possible de distributions *a posteriori* π^C que l'on peut obtenir en fonction de $\mathbf{D}_M$ construit comme un LHD maximin, nous avons besoin d'un critère quantitatif pour évaluer la proximité de π^C à la densité de probabilité cible π. Un critère connu pour ses bonnes propriétés théoriques est la divergence de Kullback-Leibler (Cover et Thomas, 1991). Calculée entre la loi *a posteriori* approchée (2.32) et la loi *a posteriori* cible (2.21), elle s'écrit

$$\mathrm{KL}(\pi(\boldsymbol{\theta}|\mathbf{z}^f)||\pi^C(\boldsymbol{\theta}|\mathbf{d})) = \int_{\mathscr{T}} \pi(\boldsymbol{\theta}|\mathbf{z}^f)\big(\log\pi(\boldsymbol{\theta}|\mathbf{z}^f) - \log\pi^C(\boldsymbol{\theta}|\mathbf{d})\big)\mathrm{d}\boldsymbol{\theta} \tag{2.34}$$

Plus la divergence de Kullback-Leibler est faible (2.34) et plus π^C est proche de π. La figure 2.10 illustre les valeurs prises par (2.34) en fonction du coefficient de prédictivité Q^2 et du RMSE de l'émulateur construit à partir d'un LHD maximin de taille M = 50. On observe que l'on ne peut pas relier directement un bon émulateur au sens d'un de ces deux critères, à une faible valeur de (2.34). En effet, pour des valeurs satisfaisantes du Q^2 (≈ 0.8) et

9. à l'aide de la librairie DiceDesign du logiciel R

FIGURE 2.8 – *Ligne pleine bleu : densité de probabilité de la distribution de probabilité approchée* $\pi^C(.|.)$ *pour plusieurs plan d'expériences LHD maximin de taille* $M = 45, 60, 75, 90$ *(de gauche à droite et de haut en bas). Ligne en tirets noirs : densité de probabilité cible* $\pi(.|.)$. *Ligne en pointillés rouges : loi a priori uniforme sur* $[5, 15]$.



du RMSE (≈ 2), (2.34) prend des valeurs comprises entre 0 et 8. Ces critères d'évaluation globale de la qualité d'un émulateur sur $\mathscr{X} \times \mathscr{T}$ ne sont par conséquent pas appropriés pour s'assurer d'un bon calage. Dans le chapitre 4, nous proposons de nouveaux algorithmes de construction séquentielle de plan d'expériences $\mathbf{D}_M$ pour rendre plus robuste le calage basé sur π^C.

2.3.3 Incertitude de prédiction

Une fois l'incertitude paramétrique *a posteriori* représentée par π si le code de calcul peut être exécuté quasi-instantanément ou par π^C si le code de calcul est coûteux, il est possible comme dans la section 2.2.2.2 de quantifier l'incertitude affectant la réponse du code de calcul. Dans le cas où π a été obtenue, un prédicteur moyen de $r(\mathbf{x})$ peut être calculé par

$$\hat{r}(\mathbf{x}) = \int y_{\boldsymbol{\theta}}(\mathbf{x})\pi(\boldsymbol{\theta}|\mathbf{d})\mathrm{d}\boldsymbol{\theta}. \tag{2.35}$$

FIGURE 2.9 – *Ligne pleine bleu : densité de probabilité de la distribution de probabilité approchée* $\pi^C(.|.)$ *pour plusieurs plans d'expériences LHD maximin de taille* $M = 50$. *Ligne en tirets noirs : densité de probabilité cible* $\pi(.|.)$. *Ligne en pointillés rouges : loi a priori uniforme sur* $[5, 15]$.

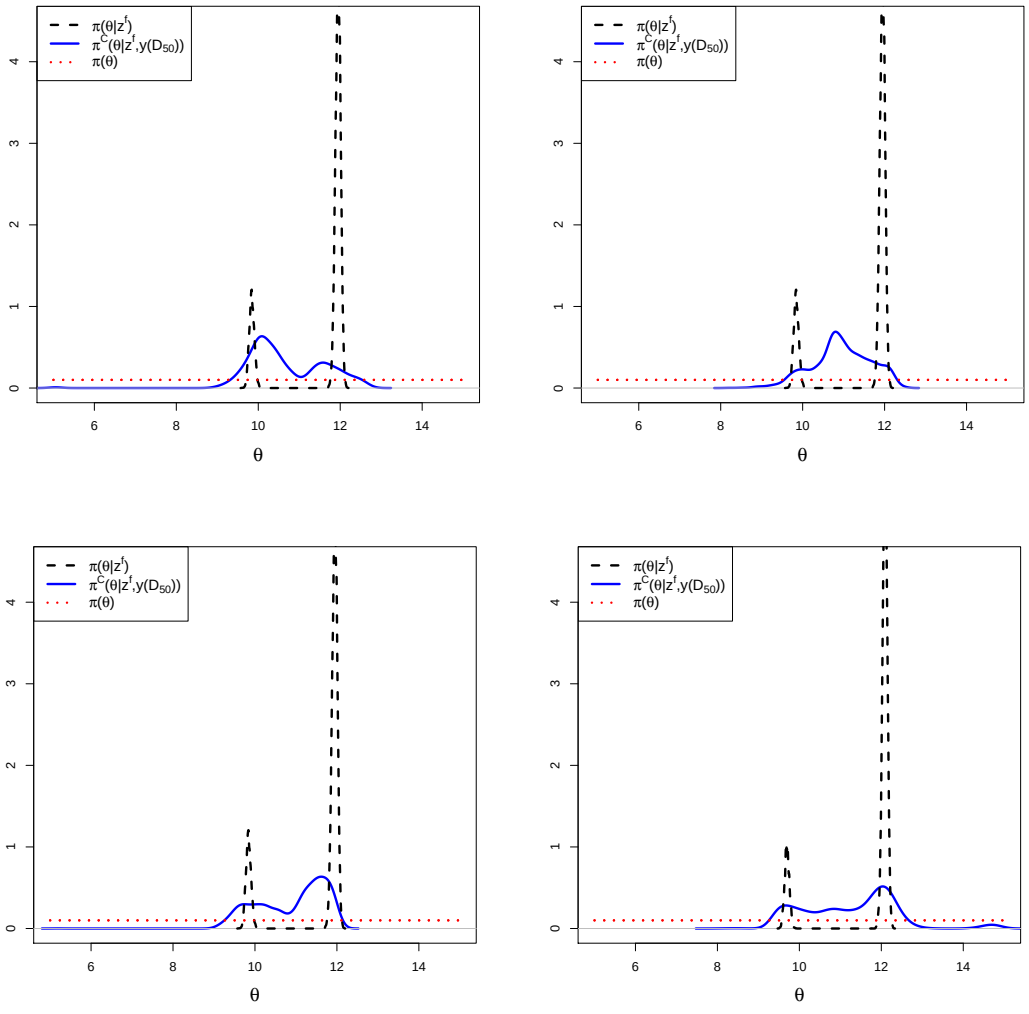
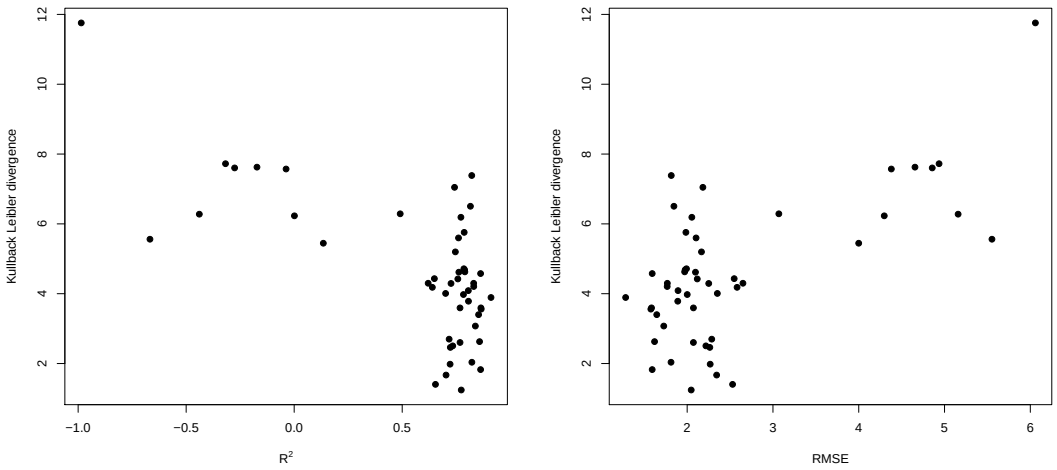


FIGURE 2.10 – *À gauche : distance de Kullback-Leibler en ordonnée (2.34) en fonction du coefficient de prédictivité* Q^2 *pour* 50 *plans d'expériences LHD maximin de taille* $M = 50$. *À droite : distance de Kullback-Leibler (2.34) en fonction du critère RMSE pour* 50 *plans d'expériences LHD maximin de taille* $M = 50$.



Le calcul d'un intervalle de crédibilité autour de $\hat{r}(\mathbf{x})$ s'appuie sur les quantiles de la loi de probabilité de $y_{\boldsymbol{\theta}}(\mathbf{x})$ calculés empiriquement à partir des échantillons $y_{\boldsymbol{\theta}_i}(\mathbf{x})$ où $\boldsymbol{\theta}_i \sim \pi(\boldsymbol{\theta}|\mathbf{d})$. La figure 2.11 illustre la loi de prédiction de la fonction (2.33). Dans le cas où les simulations sont coûteuses, un prédicteur moyen de $r(\mathbf{x})$ peut-être calculé par

$$\hat{r}(\mathbf{x}) = \int \mu_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{x}, \boldsymbol{\theta}) \pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{d}) \mathrm{d}\boldsymbol{\theta}. \tag{2.36}$$

Le calcul d'un intervalle de crédibilité autour de $\hat{r}(\mathbf{x})$ s'appuie sur les quantiles de

$$\mu_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{x}, \boldsymbol{\theta}_i) \tag{2.37}$$

où $\boldsymbol{\theta}_i \sim \pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{d})$. La figure 2.12 (à gauche) illustre le prédicteur (2.36) de la fonction jouet (2.33) obtenue à partir de la loi *a posteriori* approchée π^{C} représentée en en bas à droite de la figure 2.9. Toutefois ce prédicteur ignore l'incertitude de l'émulateur processus gaussien quantifiée par sa covariance. Ainsi, une seconde manière de construire des intervalles de crédibilité autour de (2.36) consiste à simuler des réalisations de l'émulateur processus gaussien conditionnel :

$$\mathscr{PG}\left(\mu_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(., \boldsymbol{\theta}_i), \mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}((., \boldsymbol{\theta}_i), (., \boldsymbol{\theta}_i))\right). \tag{2.38}$$

La figure 2.12 (à droite) illustre la prédiction obtenue dans ce cas. Remarquons enfin que les échantillons (2.38) sont des réalisations du processus de moyenne $\mu_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}$ égale à (2.36) et de variance $\mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}$ qui peut être calculée en utilisant la formule de la covariance totale, qui donne

$$\mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{x}) = \mathbb{E}_{\boldsymbol{\theta}}[\mathrm{V}_{\hat{\sigma}^2_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{x}, \boldsymbol{\theta})] + \mathbb{V}_{\boldsymbol{\theta}}[\mu_{\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y}(\mathbf{x}, \boldsymbol{\theta})].$$

Remarque 2.3.3.1 *En comparant la figure 2.12 à gauche et à droite, nous pouvons observer que la prise en compte de l'incertitude de l'émulateur au travers de sa covariance accroît fortement l'incertitude de prédiction, en particulier lorsque* $x \in [0, 0.5]$. *Même si la prise en*

FIGURE 2.11 – *Prédicteur moyen (2.35) sur* $[0,1]$ *(ligne pointillé) et l'enveloppe de crédibilité associée lorsque* $\mathbf{X}^f = (0.1, 0.3, 0.8)$ *et* $\lambda = 0.3$.

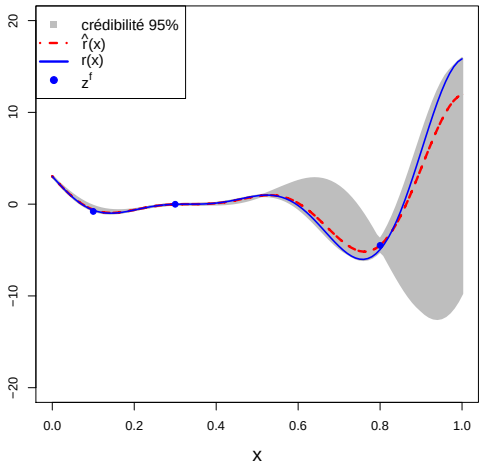
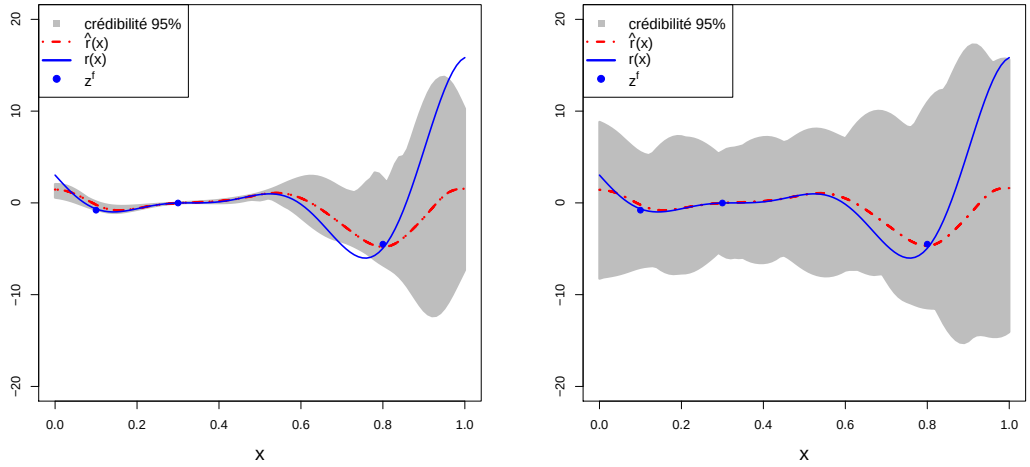


FIGURE 2.12 – *À gauche : prédiction sur* $[0,1]$ *calculée à partir de la moyenne du processus a posteriori (2.37). À droite : prédiction sur* $[0,1]$ *calculée à partir de la loi du processus a posteriori (2.38).*



compte de la covariance doit nous conduire à de plus larges plages d'incertitudes, l'écart important constaté ici s'explique par l'hypothèse de stationnarité de l'émulateur qui n'est pas vérifié par la fonction (2.33).

Dans cette section, nous avons détaillé comment valider un code de calcul en présence d'incertitude paramétrique sur $\boldsymbol{\theta}$ lorsque l'erreur de code est négligeable. Les techniques d'inférence bayésienne permettent de réaliser le calage du code, puis de calculer l'incertitude de prédiction sur le système physique $r(\mathbf{x})$ en utilisant le code ou à défaut l'émulateur processus gaussien. Toutefois, en pratique, il se peut qu'il n'existe aucune valeur du paramètre qui permette au code de représenter exactement le système physique. Dans ce cas, on parle de calage avec erreur de code, qui fait l'objet de la section suivante.

2.4 Calage des codes de calcul avec erreur de code

Il peut être plus réaliste d'aborder le problème de calage des paramètres inconnus d'un code de calcul lorsque celui ci n'est pas capable de reproduire exactement $r(\mathbf{x})$ quelle que soit la valeur de $\boldsymbol{\tau} \in \mathscr{T}$. Cette incapacité de reproduction se traduit mathématiquement de la façon suivante :

$$\forall \boldsymbol{\tau},\ \exists\, b_{\boldsymbol{\tau}}(\mathbf{x}) \in \mathrm{C}^{\star}(\mathscr{X},\mathbb{R})\ ;\ b_{\boldsymbol{\tau}}(\mathbf{x}) = r(\mathbf{x}) - y_{\boldsymbol{\tau}}(\mathbf{x}), \tag{2.39}$$

où $\mathrm{C}^{\star}(\mathscr{X},\mathbb{R})$ désigne l'espace des fonctions continues non identiquement nulles de $\mathscr{X}$ dans $\mathbb{R}$. En combinant l'équation (2.7) avec l'équation (2.39), on obtient

$$\begin{aligned} z_i^f &= r(\mathbf{x}_i^f) + \epsilon_i \\ &= y_{\boldsymbol{\tau}}(\mathbf{x}_i^f) + b_{\boldsymbol{\tau}}(\mathbf{x}_i^f) + \epsilon_i. \end{aligned} \tag{2.40}$$

Identifiabilité La fonction de vraisemblance du modèle statistique (2.40) prend des valeurs identiques pour tous les couples $\left(\boldsymbol{\tau}, b_{\boldsymbol{\tau}}(\mathbf{x})\right)$. Dans la littérature consacrée au calage de code avec erreur, une formulation alternative est proposée en considérant qu'il existe une vraie (meilleure) valeur du paramètre $\boldsymbol{\tau} = \boldsymbol{\theta}$ telle que :

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i, \tag{2.41}$$

ce qui permet de s'affranchir de la dépendance de l'erreur $b_{\boldsymbol{\tau}}(\mathbf{x})$ à la valeur de $\boldsymbol{\tau}$. Par la suite, nous discuterons plus précisément de l'interprétation à donner à $\boldsymbol{\theta}$. Bien sûr, le problème d'identifiabilité persiste car l'équation (2.41) ne diffère de l'équation (2.40) que conceptuellement. En statistique bayésienne, un tel problème n'est pas contraignant. En effet, après avoir spécifié une structure *a priori* sur l'erreur $b(\mathbf{x})$ et une loi *a priori* sur le paramètre $\boldsymbol{\theta}$ dans l'équation (2.41), les lois *a posteriori* de ces quantités peuvent être calculées à partir des données disponibles $\mathbf{d} = (\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))$. Les travaux existants modélisent l'erreur *a priori* $b(\mathbf{x})$ par un processus gaussien (voir section 2.2.2.2). Ils diffèrent cependant du point de vue des techniques d'estimations, du choix des lois *a priori* des hyperparamètres et de l'interprétation des résultats. En particulier, étant donné le problème d'identifiabilité soulevé par l'équation (2.40), on pressent que la nature du processus gaussien *a priori* sur $b(\mathbf{x})$ impacte l'estimation de $\boldsymbol{\theta}$. Nous détaillons dans les deux prochaines sections la méthode d'estimation proposée par Kennedy et O'Hagan (2001a), puis la méthode d'estimation proposée par Bayarri et al. (2007b). Nous terminons en précisant les spécificités de l'approche d'Higdon et al. (2004).

2.4.1 Méthode de Kennedy et O'Hagan

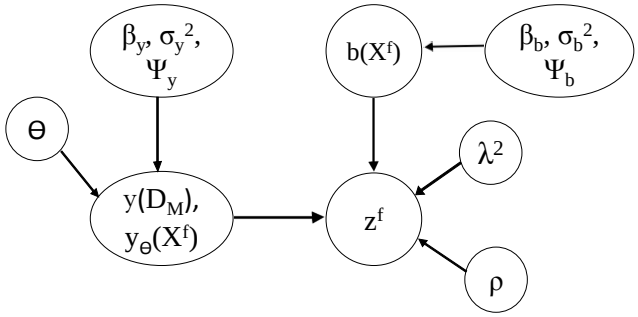
Dans ce paragraphe, nous décrivons la modélisation bayésienne hiérarchique proposée par Kennedy et O'Hagan (2001a) (KOH), qui constitue la référence fondatrice des travaux consacrés au calage d'un code de calcul coûteux. La formulation ci-dessous introduite par KOH

$$z_i^f = \rho\, y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i \tag{2.42}$$

ajoute dans l'équation (2.41) un coefficient d'échelle ρ à estimer entre les prédictions du code et la réponse physique d'intérêt $r(\mathbf{x})$. Le code de calcul et l'erreur sont modélisés *a priori* par deux processus gaussiens indépendants :

$$y(\mathbf{x},\boldsymbol{\tau}) := y_{\boldsymbol{\tau}}(\mathbf{x}) \sim \mathscr{PG}\Big(h_y(\mathbf{x},\tau)^{\mathrm{T}}\boldsymbol{\beta}_y, \sigma_y^2 \mathrm{K}_{\boldsymbol{\Psi}_y}(\mathbf{x},\boldsymbol{\tau}),(\mathbf{x}',\boldsymbol{\tau}')\Big), \tag{2.43}$$

FIGURE 2.13 – *Réseau bayésien (DAG) correspondant au modèle statistique (2.40).*



$$b(\mathbf{x}) \sim \mathscr{PG}\Big(h_b(\mathbf{x})^{\mathrm{T}}\boldsymbol{\beta}_b, \sigma_b^2 \mathrm{K}_{\boldsymbol{\Psi}_b}(\mathbf{x},\mathbf{x}')\Big).$$

Les fonctions de corrélation $\mathrm{K}_{\boldsymbol{\Psi}_y}(.,.)$ et $\mathrm{K}_{\boldsymbol{\Psi}_b}(.,.)$ utilisés par KOH sont de type exponentielles séparables :

$$\mathrm{K}_{\boldsymbol{\Psi}_y}\Big((\mathbf{x},\boldsymbol{\tau}),(\mathbf{x}',\boldsymbol{\tau}')\Big) = \exp\{-(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\Omega_x(\mathbf{x}-\mathbf{x}')\}\exp\{-(\boldsymbol{\tau}-\boldsymbol{\tau}')^{\mathrm{T}}\Omega_t(\boldsymbol{\tau}-\boldsymbol{\tau}')\},$$

$$\mathrm{K}_{\boldsymbol{\Psi}_b}\Big((\mathbf{x},\mathbf{x}')\Big) = \exp\{-(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}\Omega_x^{\star}(\mathbf{x}-\mathbf{x}')\}.$$

où Ω_x, Ω_t et $\Omega_x^{\star}$ sont des matrices diagonales. L'ensemble des quantités à estimer se compose du paramètre $\boldsymbol{\theta}$, de la variance du bruit λ^2, du paramètre d'échelle ρ et des hyperparamètres des processus gaussiens sur le code $(\boldsymbol{\beta}_y, \boldsymbol{\Psi}_y, \sigma_y^2)$ et sur l'erreur de code $(\boldsymbol{\beta}_b, \boldsymbol{\Psi}_b, \sigma_b^2)$. Notons $\boldsymbol{\beta} = (\boldsymbol{\beta}_y, \boldsymbol{\beta}_b)$ et $\mathbf{u} = (\rho, \lambda^2, \sigma_y^2, \sigma_b^2, \boldsymbol{\Psi}_y, \boldsymbol{\Psi}_b)$. D'après la formule de Bayes, la loi *a posteriori* de $(\boldsymbol{\theta}, \boldsymbol{\beta}, \mathbf{u})$ s'écrit

$$\pi(\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}|\mathbf{d}) \propto \mathscr{L}(\mathbf{d}|\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u})\pi(\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}). \tag{2.44}$$

Pour comprendre la structure hiérarchique du modèle, il peut être utile d'introduire les données latentes $\big(y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f)\big)$, puis d'écrire la vraisemblance complète par conditionnements successifs :

$$\begin{aligned}\mathscr{L}\big(\mathbf{d}, y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f)|\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}\big) = {} & \mathscr{L}(\mathbf{z}^f|\lambda^2,\rho,b(\mathbf{X}^f),y_{\boldsymbol{\theta}}(\mathbf{X}^f))\mathscr{L}(b(\mathbf{X}^f)|\boldsymbol{\beta}_b,\boldsymbol{\Psi}_b,\sigma_b^2) \\ & \mathscr{L}(y_{\boldsymbol{\theta}}(\mathbf{X}^f)|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\beta}_y,\boldsymbol{\Psi}_y,\sigma_y^2,\boldsymbol{\theta})\mathscr{L}(y(\mathbf{D}_{\mathrm{M}})|\boldsymbol{\beta}_y,\boldsymbol{\Psi}_y,\sigma_y^2)\end{aligned} \tag{2.45}$$

avec

$$\begin{aligned}\mathbf{z}^f|\lambda^2,\rho,b(\mathbf{X}^f),y_{\boldsymbol{\theta}}(\mathbf{X}^f) &\sim \mathscr{N}(\rho\, y_{\boldsymbol{\theta}}(\mathbf{X}^f) + b(\mathbf{X}^f), \lambda^2\mathbf{I}_n),\\ b(\mathbf{X}^f)|\sigma_b^2,\boldsymbol{\beta}_b,\boldsymbol{\Psi}_b &\sim \mathscr{N}(h_b(\mathbf{X}^f)^{\mathrm{T}}\boldsymbol{\beta}_b, \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}(\mathbf{X}^f)),\\ y_{\boldsymbol{\theta}}(\mathbf{X}^f)|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\beta}_y,\boldsymbol{\theta},\boldsymbol{\Psi}_y &\sim \mathscr{N}(\mu,\sigma_y^2\Sigma),\\ y(\mathbf{D}_{\mathrm{M}})|\boldsymbol{\beta}_y,\sigma_y^2,\boldsymbol{\Psi}_y &\sim \mathscr{N}(h_y(\mathbf{D}_{\mathrm{M}})^{\mathrm{T}}\boldsymbol{\beta}_y, \sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}})),\end{aligned}$$

où

$$\mu = h_y(\mathbf{D}_{\boldsymbol{\theta}})^{\mathrm{T}}\boldsymbol{\beta}_y + \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}})\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}})^{-1}(y(\mathbf{D}_{\mathrm{M}}) - h_y(\mathbf{D}_{\mathrm{M}})\boldsymbol{\beta}_y),$$

et

$$\Sigma = \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\boldsymbol{\theta}}) - \Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}})\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}})^{-1}\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\boldsymbol{\theta}},\mathbf{D}_{\mathrm{M}}).$$

La figure 2.13 illustre le DAG qui découle de l'équation (2.45). La vraisemblance des données observées peut être calculée en marginalisant la vraisemblance complète par rapport à la distribution des données latentes :

$$\begin{aligned}\mathscr{L}(\mathbf{d}|\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}) &= \int \mathscr{L}\big(\mathbf{d}, y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f)|\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}\big)\mathrm{d}y_{\boldsymbol{\theta}}(\mathbf{X}^f)\mathrm{d}b(\mathbf{X}^f)\\ &= \mathscr{L}(y(\mathbf{D}_{\mathrm{M}})|\boldsymbol{\beta}_y,\boldsymbol{\Psi}_y,\sigma_y^2)\mathscr{L}(\mathbf{z}^f|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\theta},\boldsymbol{\beta},\sigma_y^2,\sigma_b^2,\boldsymbol{\Psi}_y,\boldsymbol{\Psi}_b,\lambda^2).\end{aligned} \tag{2.46}$$

On obtient

$$\mathbf{z}^f|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\theta},\boldsymbol{\Psi}_y,\boldsymbol{\Psi}_b,\lambda^2 \sim \mathscr{N}(\mu + h_b(\mathbf{X}^f)\beta_b, \sigma_y^2\Sigma + \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}(\mathbf{X}^f) + \lambda^2\mathbf{I}_n), \tag{2.47}$$

et donc (2.46) s'écrit comme le produit de deux densités de probabilité gaussiennes. On peut obtenir l'expression de la vraisemblance à une constante de proportionnalité près :

$$\mathscr{L}(\mathbf{d}|\boldsymbol{\theta},\boldsymbol{\beta},\mathbf{u}) \propto |\mathrm{V}_d(\boldsymbol{\theta})|^{-1/2}\exp -\frac{1}{2}\big(\mathbf{d}-\mathrm{H}(\boldsymbol{\theta})\boldsymbol{\beta}\big)^{\mathrm{T}}\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}\big(\mathbf{d}-\mathrm{H}(\boldsymbol{\theta})\boldsymbol{\beta}\big), \tag{2.48}$$

avec

$$\mathrm{H}(\boldsymbol{\theta}) = \begin{pmatrix} h_y(\mathbf{D}_{\mathrm{M}})^{\mathrm{T}} & 0 \\ \rho h_y(\mathbf{D}_{\boldsymbol{\theta}})^{\mathrm{T}} & h_b(\mathbf{X}^f)^{\mathrm{T}}\end{pmatrix}, \tag{2.49}$$

et

$$\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta}) = \begin{pmatrix} \sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}}) & \rho\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}\big(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}\big)^{\mathrm{T}} \\ \rho\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}\big(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}\big) & \lambda^2\mathbf{I}_n + \rho^2\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}) + \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}(\mathbf{z}^f)\end{pmatrix}. \tag{2.50}$$

En considérant la loi *a priori* de Jeffreys $\pi(\boldsymbol{\beta}) \propto 1$ pour $\boldsymbol{\beta}$, l'intégration de (2.44) relativement à $\boldsymbol{\beta}$ est analytique. On obtient,

$$\pi(\boldsymbol{\theta},\mathbf{u}|\mathbf{d}) \propto |\mathrm{V}_{\mathbf{d}}(\theta))|^{\frac{1}{2}}\exp -\frac{1}{2}(\mathbf{d}-\mathrm{H}(\boldsymbol{\theta})\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}(\mathbf{d}-\mathrm{H}(\boldsymbol{\theta})\hat{\boldsymbol{\beta}})\pi(\boldsymbol{\theta})\pi(\mathbf{u}), \tag{2.51}$$

avec

$$\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) = \mathrm{W}(\boldsymbol{\theta})\mathrm{H}(\boldsymbol{\theta})^{\mathrm{T}}\mathrm{V}_{\mathrm{D}}^{-1}(\boldsymbol{\theta})\mathbf{d}$$

et

$$\mathrm{W}(\boldsymbol{\theta}) = \mathrm{H}(\boldsymbol{\theta})^{\mathrm{T}}\mathrm{V}_{\mathrm{D}}^{-1}(\boldsymbol{\theta})\mathrm{H}(\boldsymbol{\theta}).$$

De plus,

$$\boldsymbol{\beta}|\mathbf{d},\mathbf{u},\boldsymbol{\theta} \sim \mathscr{N}(\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}),\mathrm{W}(\boldsymbol{\theta})). \tag{2.52}$$

Le calcul de $\pi(\boldsymbol{\theta}|\mathbf{d})$ nécessite d'intégrer (2.51) par rapport à $\mathbf{u}$. Toutefois, au regard de la dimension de $\mathbf{u}$, les méthodes de quadratures ne peuvent pas être envisagées. En effet, même en considérant une fonction de covariance isotrope, la dimension de $\mathbf{u}$ est au minimum 6. C'est pourquoi, Kennedy et O'Hagan (2001a) proposent une estimation en deux étapes du couple $(\boldsymbol{\theta},\mathbf{u})$. En suivant la même idée que pour le calage sans erreur, les hyperparamètres du processus gaussien sur le code $(\sigma_y^2,\boldsymbol{\Psi}_y)$ sont d'abord estimés en maximisant la vraisemblance partielle des simulations $y(\mathbf{D}_{\mathrm{M}})$, ce qui nous donne un estimateur $(\hat{\sigma}_y^2,\hat{\boldsymbol{\Psi}}_y)$, puis $(\sigma_b^2,\boldsymbol{\Psi}_b,\rho,\lambda)$ sont estimés en maximisant la vraisemblance conditionnelle de $\mathbf{z}^f$ par rapport aux simulations $y(\mathbf{D}_{\mathrm{M}})$ intégrée sur la loi *a priori* de $\boldsymbol{\theta}$. Cette vraisemblance n'est pas gaussienne mais est approchée comme telle pour le calcul d'un estimateur $(\hat{\sigma}_b^2,\hat{\boldsymbol{\Psi}}_b,\hat{\rho},\hat{\lambda})$ en maximisant,

$$\mathbf{z}^f|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2 \sim \mathscr{N}\big(\mathbb{E}[\mathbf{z}^f|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2],\mathbb{V}[\mathbf{z}^f|y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2]\big) \tag{2.53}$$

où

$$\mathbb{E}[\mathbf{z}^f|y(\mathbf{D}_\mathrm{M}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2] = \int \mathbb{E}[\mathbf{z}^f|y(\mathbf{D}_\mathrm{M}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2,\boldsymbol{\theta}]\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{2.54}$$

et

$$\mathbb{V}[\mathbf{z}^f|y(\mathbf{D}_\mathrm{M}),\boldsymbol{\Psi}_b,\boldsymbol{\beta}_b,\rho,\lambda^2] = \rho^2\mathrm{C} + \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}(\mathbf{X}^f) + \lambda^2\mathrm{I}_n. \tag{2.55}$$

Dans (2.55), l'élément (i,j) de la matrice C s'écrit pour $1 \leq i,j \leq n$

$$\int Cov\big(y(\mathbf{x}_i,\boldsymbol{\theta}),y(\mathbf{x}_j,\boldsymbol{\theta}))|y(\mathbf{D}_\mathrm{M}),\boldsymbol{\theta},\hat{\sigma}_y^2,\hat{\boldsymbol{\Psi}}_y\big)\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}. \tag{2.56}$$

En choisissant une distribution gaussienne *a priori* pour $\boldsymbol{\theta}$

$$\boldsymbol{\theta} \sim \mathcal{N}(\bar{\boldsymbol{\theta}},\sigma_{\boldsymbol{\theta}}^2\mathrm{I}) \tag{2.57}$$

les expressions (2.54) et (2.55) sont analytiques. Pour le détail des calculs, se reporter à (Kennedy et O'Hagan, 2001b). Finalement, l'estimateur ponctuel $\hat{\mathbf{u}} := (\hat{\rho},\hat{\lambda}^2,\hat{\sigma}_y^2,\hat{\sigma}_b^2,\hat{\boldsymbol{\Psi}}_y,\hat{\boldsymbol{\Psi}}_b)$ est injecté dans la densité *a posteriori* (2.51), à partir de laquelle la loi *a posteriori* de $\boldsymbol{\theta}$ est échantillonnée via un algorithme de Métropolis-Hastings. Après avoir calculé $\pi(\boldsymbol{\theta}|\mathbf{d},\hat{\mathbf{u}})$, des prédictions de la quantité physique $r(\mathbf{x})$ peuvent être calculées en utilisant la distribution $r(\mathbf{x})|\mathbf{d},\boldsymbol{\theta},\mathbf{u}$ qui est gaussienne de moyenne

$$\mathbb{E}[r(\mathbf{x})|\mathbf{d},\boldsymbol{\theta},\hat{\mathbf{u}}] = h(\mathbf{x},\boldsymbol{\theta})^\mathrm{T}\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) + t(\mathbf{x},\boldsymbol{\theta})\mathrm{V}_d(\boldsymbol{\theta})^{-1}(\mathbf{d}-\mathrm{H}(\boldsymbol{\theta})\hat{\boldsymbol{\beta}}(\boldsymbol{\theta}) \tag{2.58}$$

avec

$$h(\mathbf{x},\boldsymbol{\theta}) = \begin{pmatrix}\rho h_y(\mathbf{x},\boldsymbol{\theta})\\ h_b(\mathbf{x})\end{pmatrix} \text{ et } t(\mathbf{x},\boldsymbol{\theta}) = \begin{pmatrix}\rho\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}((\mathbf{x},\boldsymbol{\theta}),\mathbf{D}_\mathrm{M})\\ \rho^2\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}((\mathbf{x},\boldsymbol{\theta}),\mathbf{D}_{\boldsymbol{\theta}}) + \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}(\mathbf{x},\mathbf{X}^f)\end{pmatrix}$$

et de matrice de variance-covariance

$$\begin{aligned}Cov\big(r(\mathbf{x}_1),r(\mathbf{x}_2)|\mathbf{d},\boldsymbol{\theta},\hat{\mathbf{u}}\big) = {}& \rho^2\sigma_y^2\Sigma_{\boldsymbol{\Psi}_y}\big((\mathbf{x}_1,\boldsymbol{\theta}),(\mathbf{x}_2,\boldsymbol{\theta})\big)+\\ & \sigma_b^2\Sigma_{\boldsymbol{\Psi}_b}\big(\mathbf{x}_1,\mathbf{x}_2\big) - t(\mathbf{x}_1,\boldsymbol{\theta})^\mathrm{T}\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}t(\mathbf{x}_2,\boldsymbol{\theta})+\\ & \big(h(\mathbf{x}_1,\boldsymbol{\theta}) - \mathrm{H}(\boldsymbol{\theta})\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}t(\mathbf{x}_1,\boldsymbol{\theta})\big)^\mathrm{T}\mathrm{W}(\boldsymbol{\theta})\big(h(\mathbf{x}_2,\boldsymbol{\theta}) - \mathrm{H}(\boldsymbol{\theta})\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}t(\mathbf{x}_2,\boldsymbol{\theta})\big)-\\ & \mathrm{H}(\boldsymbol{\theta})^\mathrm{T}\mathrm{V}_{\mathbf{d}}(\boldsymbol{\theta})^{-1}t(\mathbf{x}_2,\boldsymbol{\theta}).\end{aligned} \tag{2.59}$$

Le prédicteur que préconisent KOH intègre l'incertitude *a posteriori* sur $\boldsymbol{\theta}$. Il est donné par,

$$\hat{r}(\mathbf{x}) = \int \mathbb{E}[r(\mathbf{x})|\mathbf{d},\boldsymbol{\theta},\hat{\mathbf{u}}]\pi(\boldsymbol{\theta}|\mathbf{d},\mathbf{u})\mathrm{d}\boldsymbol{\theta}. \tag{2.60}$$

La calcul d'un intervalle de crédibilité autour de $\hat{r}(\mathbf{x})$ s'appuie sur les quantiles de la moyenne du processus *a posteriori* $\mathbb{E}[r(\mathbf{x})|\mathbf{d},\boldsymbol{\theta},\hat{\mathbf{u}}])$ calculés empiriquement à partir d'échantillons $\mathbb{E}[r(\mathbf{x})|\mathbf{d},\boldsymbol{\theta_i},\hat{\mathbf{u}}])$ où $\boldsymbol{\theta_i} \sim \pi(\boldsymbol{\theta}|\mathbf{d},\mathbf{u})$. Ce prédicteur ignore donc l'incertitude portée par la covariance (2.59). L'exemple académique que nous étudions dans la section suivante illustre l'importance de prendre en compte toute la distribution du processus *a posteriori* pour construire les intervalles de crédibilité autour de $\hat{r}(\mathbf{x})$ à partir des échantillons

$$r(.|\boldsymbol{\theta}_i) \sim \mathcal{PG}\Big(\mathbb{E}[r(.)|\mathbf{d},\boldsymbol{\theta_i},\hat{\mathbf{u}}],Cov\big(r(.),r(.)|\mathbf{d},\boldsymbol{\theta_i},\hat{\mathbf{u}}\big)\Big). \tag{2.61}$$

2.4.1.1 Exemple d'application à une fonction analytique

Dans ce paragraphe, nous appliquons la technique d'estimation décrite précédemment à la fonction analytique ci-dessous jouant le rôle du code de calcul :

$$y_\tau : x \longrightarrow (6x-2)^2 \sin(\tau x - 4) \tag{2.62}$$

et

$$r : x \longrightarrow (6x-2)^2 \sin(12x-4) + 10(x-0.5) - 5.$$

En prenant $n = 4$, $n = 6$, puis $n = 8$ avec à chaque fois $\lambda = 0.3$, les données $\mathbf{z}^f$ sont simulées suivant l'équation (2.40). La loi *a priori* $\pi(\boldsymbol{\theta})$ que nous choisissons est gaussienne de moyenne nulle et de variance $\sigma_\theta^2 = 100$, donc peu informative. Nous avons utilisé la librairie R BACCO qui regroupe l'ensemble des fonctions permettant d'effectuer le calage d'un code de calcul coûteux selon la méthode proposée par KOH (Hankin, 2005). Sa principale difficulté d'utilisation réside en la spécification des fonctions de tendance $h_y(.)$ et $h_b(.)$. En effet, l'utilisateur doit réécrire les fonctions correspondantes *h1.int* et *h2.int* qui implémentent ses choix de tendances, puis réécrire en conséquence la fonction *E.theta.int* qui calcule les expressions

$$\mathbb{E}_\theta[h_y(\mathbf{x},\theta)] = \int h_y(\mathbf{x},\theta)\pi(\theta)\mathrm{d}\theta$$

et

$$Cov(h_y(x_1,\theta), h_y(x_2,\theta)) = \mathbb{E}[h_y(x_1,\theta)h_y(x_2,\theta)^{\mathrm{T}}] - \mathbb{E}[h_y(x_1,\theta)]\mathbb{E}[h_y(x_2,\theta)]^{\mathrm{T}}$$

qui sont utilisées dans la procédure d'estimation de KOH. Dans cet exemple, nous avons choisi une tendance polynomiale d'ordre 2 pour le processus gaussien sur le code

$$h_y(x,\theta) = (1, x, \theta, \theta x, x^2, \theta^2)^{\mathrm{T}}. \tag{2.63}$$

D'après (2.63),

$$\mathbb{E}_\theta[h_y(x,\theta)] = (1, x, \bar{\theta}, \bar{\theta}x, x^2, \bar{\theta}^2 + \sigma_\theta^2)^{\mathrm{T}} \tag{2.64}$$

et

$$Cov(h_y(x_1,\theta), h_y(x_2,\theta)) := \mathbb{E}[h_y(x_1,\theta)h_y(x_2,\theta)^{\mathrm{T}}] - \mathbb{E}[h_y(x_1,\theta)]\mathbb{E}[h_y(x_2,\theta)]^{\mathrm{T}}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & \sigma_\theta^2 x_1 & 0 & \overline{\theta^3} - \bar{\theta}^3 - \bar{\theta}\sigma_\theta^2 \\ 0 & 0 & \sigma_\theta^2 x_2 & \sigma_\theta^2 x_1 x_2 & 0 & (\overline{\theta^3} - \bar{\theta}^3 - \bar{\theta}\sigma_\theta^2)x_2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{\theta^3} - \bar{\theta}^3 - \bar{\theta}\sigma_\theta^2 & (\overline{\theta^3} - \bar{\theta}^3 - \bar{\theta}\sigma_\theta^2)x_1 & 0 & \overline{\theta^4} - (\bar{\theta}^2 + \sigma_\theta^2)^2 \end{pmatrix}$$

où $\overline{\theta^3}$ et $\overline{\theta^4}$ sont les notations pour les moments d'ordre 3 et 4 de la loi normale *a priori* sur θ. D'après (2.57), $\bar{\theta} = 0$, ce qui implique $\overline{\theta^3} = 0$ et $\overline{\theta^4} = 3\sigma_\theta^4$. Par suite,

$$Cov(h_y(x_1,\theta), h_y(x_2,\theta)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 & \sigma_\theta^2 x_1 & 0 & 0 \\ 0 & 0 & \sigma_\theta^2 x_2 & \sigma_\theta^2 x_1 x_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3\sigma_\theta^4 \end{pmatrix}.$$

FIGURE 2.14 – *Méthode de KOH. À gauche : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = \{0, 0.4, 0.6, 1\}$. *À droite : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$. *En bas : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$. *Les processus gaussiens a priori sont choisis avec une tendance quadratique sur le code* $h_y(x, \theta) = (1, x, \theta, \theta x, x^2, \theta^2)^{\mathrm{T}}$ *et une tendance linéaire sur l'erreur de code* $h_b(x) = (1, x)^{\mathrm{T}}$. *Le processus gaussien sur le code est construit à partir d'un plan d'expériences en grille de taille* $\mathrm{M} = 90$ *sur* $[0, 1] \times [5, 15]$.

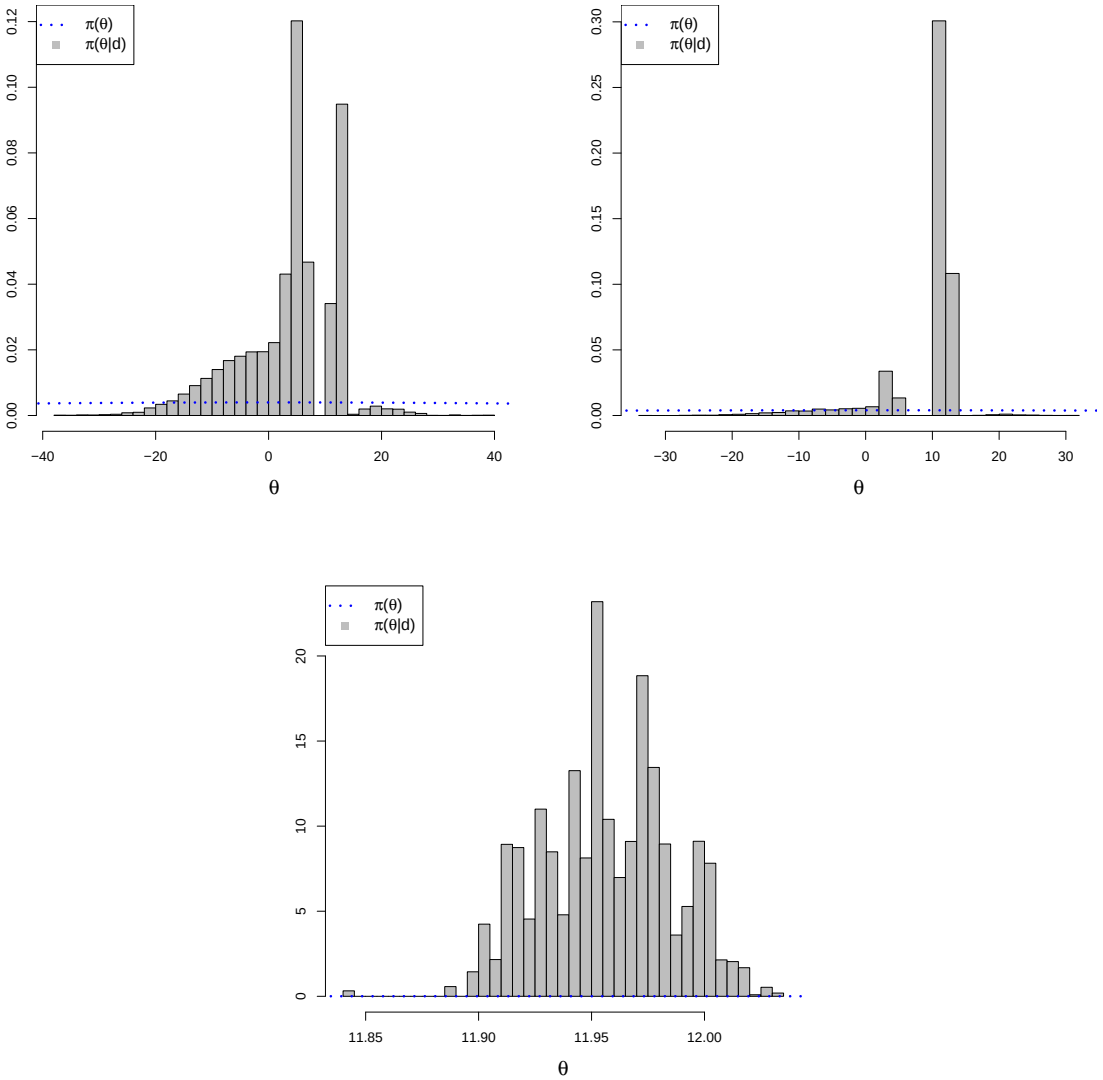
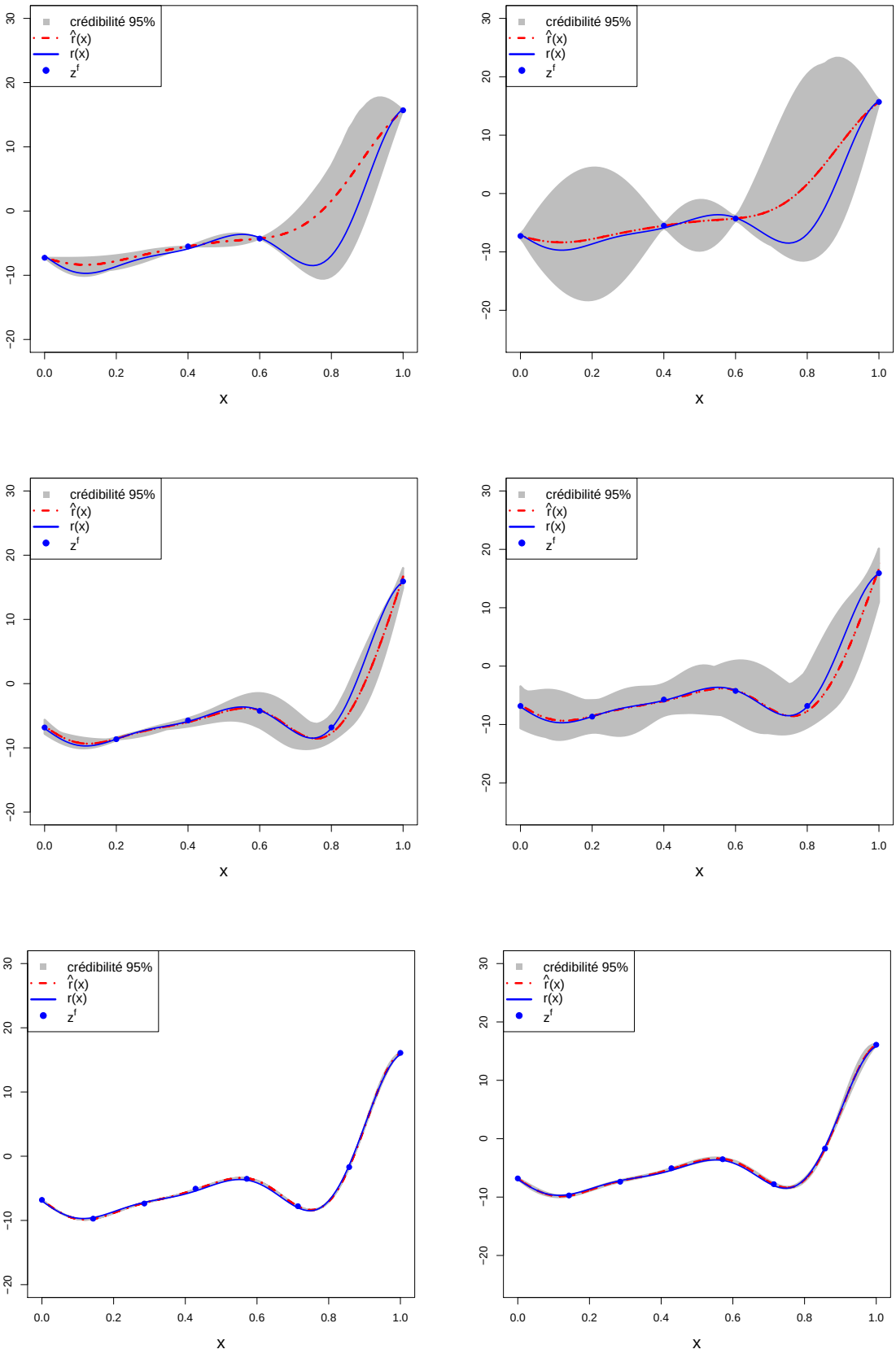


FIGURE 2.15 – *Méthode de KOH. De haut en bas : prédiction de* $r(x)$ *sur* $[0,1]$ *lorsque* $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$, $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$ *et* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$. *Les processus gaussiens a priori sont choisis avec une tendance quadratique sur le code* $h_y(x, \theta) = (1, x, \theta, \theta x, x^2, \theta^2)^{\mathrm{T}}$ *et une tendance linéaire sur l'erreur de code* $h_b(x) = (1, x)^{\mathrm{T}}$. *À gauche : prédiction calculée à partir de la moyenne du processus a posteriori (2.60). À droite : prédiction calculée à partir de la loi du processus a posteriori (2.61). Le processus gaussien sur le code est construit à partir d'un plan d'expériences en grille de taille* $\mathrm{M} = 90$ *sur* $[0,1] \times [5,15]$.



Remarque 2.4.1.1 *Le choix d'une densité gaussienne pour* $\pi(\boldsymbol{\theta})$ *permet de calculer analytiquement les intégrales (2.54) et (2.55) qui interviennent dans la procédure d'estimation proposée par KOH. Toutefois, le fait que cette loi soit à support infini est en contradiction avec la construction de l'émulateur qui s'effectue sur l'espace bornée* $\mathscr{X} \times \mathscr{T} = [0,1] \times [5,15]$. *En effet, l'émulateur processus gaussien que l'on construit n'est pas capable de fournir de bonnes prédictions du code en extrapolation, c'est à dire sur* $\mathbb{R} \backslash [5,15]$ *(Brynjarsdottir et O'Hagan, 2014). Par suite, le support de* $\pi(\boldsymbol{\theta})$ *devrait logiquement se restreindre à* $\mathscr{T}$.

Avec une tendance linéaire $\mathbf{h_b}(\mathbf{x}) = (\mathbf{1}, \mathbf{x})$ *a priori* pour $\mathbf{b}(\mathbf{x})$ Le choix de la tendance linéaire sur l'erreur va rendre plus probable les éventuelles valeurs de τ qui correspondent à une erreur linéaire $b_\tau(x)$. Dans cet exemple, c'est le cas pour la valeur $\tau = 12$ qui définie l'erreur linéaire

$$b_{\tau=12}(x) = 10(x - 0.5) - 5.$$

On remarque en effet que $\pi(\theta|\mathbf{d})$ admet comme maximum une valeur θ de plus en plus proche de $\theta = 12$ lorsque le nombre de mesures n augmente comme l'illustre la figure 2.14. La figure 2.15 illustre les lois de prédictions de $r(x)$ sur $[0,1]$ lorsque $n = 4$, $n = 6$ et $n = 8$. Comme attendu, on observe une réduction de l'incertitude sur $\hat{r}(x)$ lorsque n augmente jusqu'à devenir quasi-nulle pour $n = 8$.

Avec une tendance $\mathbf{h_b}(\mathbf{x}) = (\mathbf{1}, \mathbf{x^4})$ *a priori* pour $\mathbf{b}(\mathbf{x})$ La figure 2.16 illustre la loi *a posteriori* lorsque $n = 4$, $n = 6$ et $n = 8$. Aucune valeur de τ ne définit une erreur $b_\tau(\mathbf{x})$ de tendance $h_b(x) = (1, x^4)$, ce qui peut expliquer l'allure multimodale de $\pi(\theta|\mathbf{d})$. La figure 2.17 illustre les lois de prédiction de $r(\mathbf{x})$ correspondantes. Comme pour la tendance linéaire, l'incertitude de prédiction diminue quand n augmente même si l'on observe que les intervalles de crédibilité sont un peu plus larges.

Conclusion La diminution de l'incertitude de prédiction reste relativement robuste à la spécification du processus gaussien *a priori* sur $b(x)$, ce qui avait déjà été observé dans les travaux de Bayarri et al. (2007b) que nous présentons dans la prochaine section. Toutefois, nous avons mis en évidence l'importance d'utiliser à la fois la moyenne et la variance du processus *a posteriori* sur $r(x)$ pour construire les intervalles de crédibilité. En effet, la figure 2.17 (en haut à gauche) montre un intervalle de crédibilité à 95% qui ne contient pas la réponse physique pour $x \in [0.6, 0.85]$!

2.4.2 Méthode de Bayarri

La modélisation décrite dans Bayarri et al. (2007b) est similaire à celle présentée par KOH, hormis le paramètre d'échelle ρ qui est fixé à 1. L'équation (2.42) se réécrit donc

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i. \tag{2.65}$$

La méthode combine l'estimation de certains hyperparamètres par maximum de vraisemblance avec l'échantillonnage des lois *a posteriori* de $\boldsymbol{\theta}$ et des hyperparamètres restants. Comme dans Kennedy et O'Hagan (2001a), un estimateur ponctuel $(\hat{\boldsymbol{\beta}}_y, \hat{\boldsymbol{\Psi}}_y, \hat{\sigma}_y^2)$ est d'abord calculé en maximisant la vraisemblance de $y(\mathbf{D}_\mathrm{M})$, puis un estimateur ponctuel $(\hat{\boldsymbol{\beta}}_b, \hat{\boldsymbol{\Psi}}_b, \hat{\sigma}_b^2, \hat{\lambda})$ est calculé en maximisant la vraisemblance des erreurs virtuelles $\mathbf{z}^f - y(\mathbf{D}_{\boldsymbol{\theta}^\star})$ simulées à partir d'une meilleure valeur *a priori* $\boldsymbol{\theta}^\star$ que l'on croit la plus proche possible de $\boldsymbol{\theta}$. Dans la suite du processus d'inférence, les estimateurs $\hat{\sigma}_b^2$ et $\hat{\lambda}^2$ sont utilisés pour caler les lois *a priori* $\pi(\sigma_b^2)$ et $\pi(\lambda^2)$ inverse-gamma tandis que les hyperparamètres $\boldsymbol{\beta}_b$ et

FIGURE 2.16 – *Méthode de KOH. À gauche : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$. *À droite : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$. *En bas : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$. *Les processus gaussiens a priori sont choisis avec une tendance quadratique sur le code* $h_y(x, \theta) = (1, x, \theta, \theta x, x^2, \theta^2)^{\mathrm{T}}$ *et une tendance linéaire* $h_b(x) = (1, x^4)^{\mathrm{T}}$ *sur l'erreur de code. Le processus gaussien sur le code est construit à partir d'un plan d'expériences en grille de taille* $\mathrm{M} = 90$ *sur* $[0, 1] \times [5, 15]$.

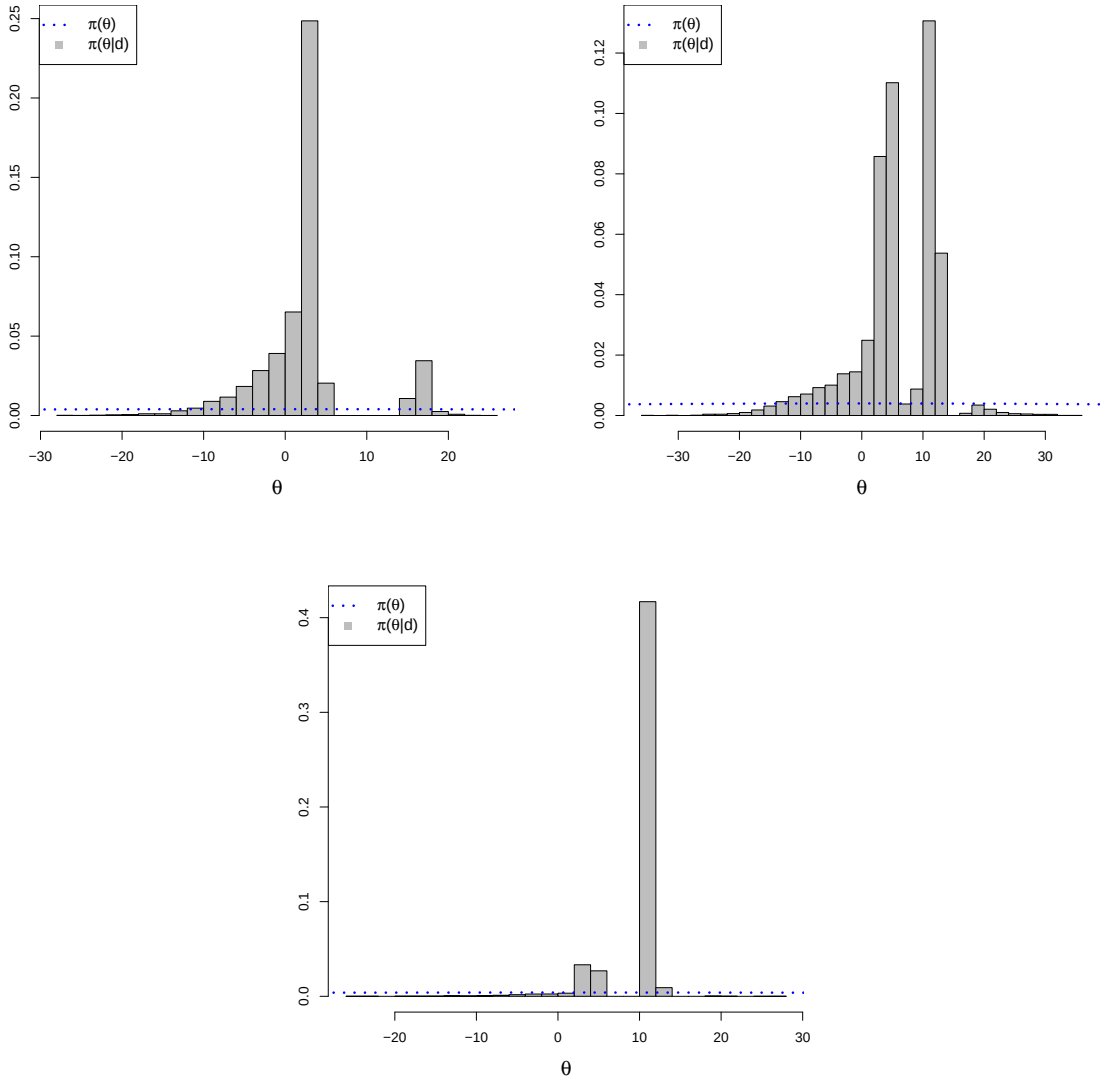
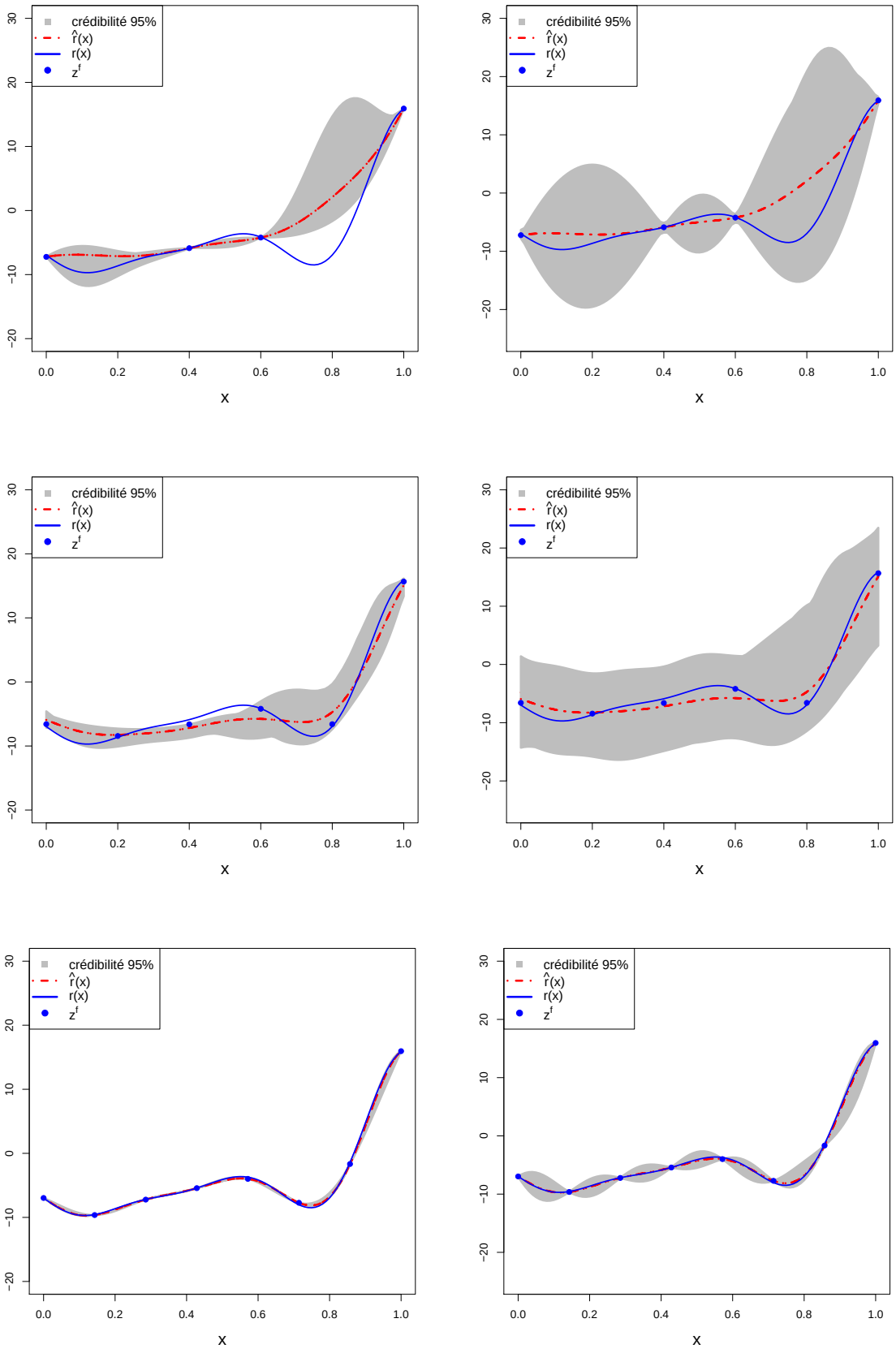


FIGURE 2.17 – *Méthode de KOH. De haut en bas : prédiction de* $r(x)$ *sur* $[0,1]$ *lorsque* $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$, $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$ *et* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$. *Les processus gaussiens a priori sont choisis avec une tendance quadratique sur le code* $h_y(x, \theta) = (1, x, \theta, \theta x, x^2, \theta^2)^{\mathrm{T}}$ *et une tendance linéaire* $h_b(x) = (1, x^4)^{\mathrm{T}}$ *sur l'erreur de code. À gauche : prédiction calculée à partir de la moyenne du processus a posteriori (2.60). À droite : prédiction calculée à partir de la loi du processus a posteriori (2.61). Le processus gaussien sur le code est construit à partir d'un plan d'expériences en grille de taille* $\mathrm{M} = 90$ *sur* $[0,1] \times [5,15]$.



$\boldsymbol{\Psi}_b$ sont fixés respectivement à $\hat{\boldsymbol{\beta}}_b$ et $\hat{\boldsymbol{\Psi}}_b$. L'échantillonnage des lois *a posteriori* de σ_b^2, λ et $\boldsymbol{\theta}$ s'effectue grâce à un algorithme de Gibbs basé sur les lois conditionnelles complètes ci-dessous :

$$y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f)|\boldsymbol{\theta}, \sigma_b^2, \lambda^2, \mathbf{d} \sim \mathcal{N}(,), \tag{2.66}$$

$$\lambda^2|y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f), \boldsymbol{\theta}, \sigma_b^2, \mathbf{d} \sim \mathcal{IG}(.,.), \tag{2.67}$$

$$\sigma_b^2|y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f), \boldsymbol{\theta}, \lambda^2, \mathbf{d} \sim \mathcal{IG}(.,.), \tag{2.68}$$

$$\boldsymbol{\theta}|y_{\boldsymbol{\theta}}(\mathbf{X}^f), b(\mathbf{X}^f), \sigma_b^2, \lambda^2, \mathbf{d} \sim \mathcal{L}(.)\pi(\boldsymbol{\theta}) \tag{2.69}$$

Les paramètres des distributions (2.66), (2.67) et (2.68) sont rapportés dans Bayarri et al. (2007b). L'échantillonnage de $\boldsymbol{\theta}$ à partir de (2.69) est effectué grâce à un algorithme de Metropolis-Hastings. Prédire la réponse physique $r(\mathbf{x})$ en une nouvelle configuration d'entrée $\mathbf{x}$ en tenant compte de l'incertitude *a posteriori* sur $\mathbf{u}$ revient à échantillonner la loi prédictive *a posteriori* donnée par

$$\int \pi(y_{\boldsymbol{\theta}}(\mathbf{x}), b(\mathbf{x})|\mathbf{d}, \mathbf{u}, \boldsymbol{\theta})\pi(\mathbf{u}, \boldsymbol{\theta}|\mathbf{d})\mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta} \tag{2.70}$$

où $\mathbf{u} = (\lambda^2, \boldsymbol{\theta}, \sigma_b^2)$. À partir d'un nombre suffisamment grand $y^i(\mathbf{x})$ et $b^i(\mathbf{x})$ issus de (2.70), des prédicteurs de $b(\mathbf{x})$ et $r(\mathbf{x})$ peuvent être calculés par

$$\hat{b}(\mathbf{x}) = \sum_{i=1}^{N} b^i(\mathbf{x}) \tag{2.71}$$

et

$$\hat{r}(\mathbf{x}) = \sum_{i=1}^{N} y^i(\mathbf{x}) + b^i(\mathbf{x}). \tag{2.72}$$

Ces échantillons permettent aussi le calcul d'intervalles de crédibilité autour de ces prédicteurs.

2.4.2.1 Exemple d'application à une fonction analytique

Nous appliquons la méthode de Bayarri et al. (2007b) au même exemple que pour l'illustration de la méthode de KOH :

$$y_\tau : x \longrightarrow (6x-2)^2 \sin(\tau x - 4) \tag{2.73}$$

et

$$r : x \longrightarrow (6x-2)^2 \sin(12x-4) + 10(x-0.5) - 5.$$

La figure 2.18 illustre la loi *a posteriori* $\pi(\theta|\mathbf{d})$ lorsque $n = 4$, $n = 6$ et $n = 8$. Comme dans la méthode de KOH, le mode de cette distribution est situé autour de la valeur $\theta = 12$ lorsque n augmente. La figure 2.19 illustre les prédictions de $r(x)$ sur $[0,1]$. Comme attendu, l'incertitude autour du prédicteur (2.72) diminue lorsque n augmente.

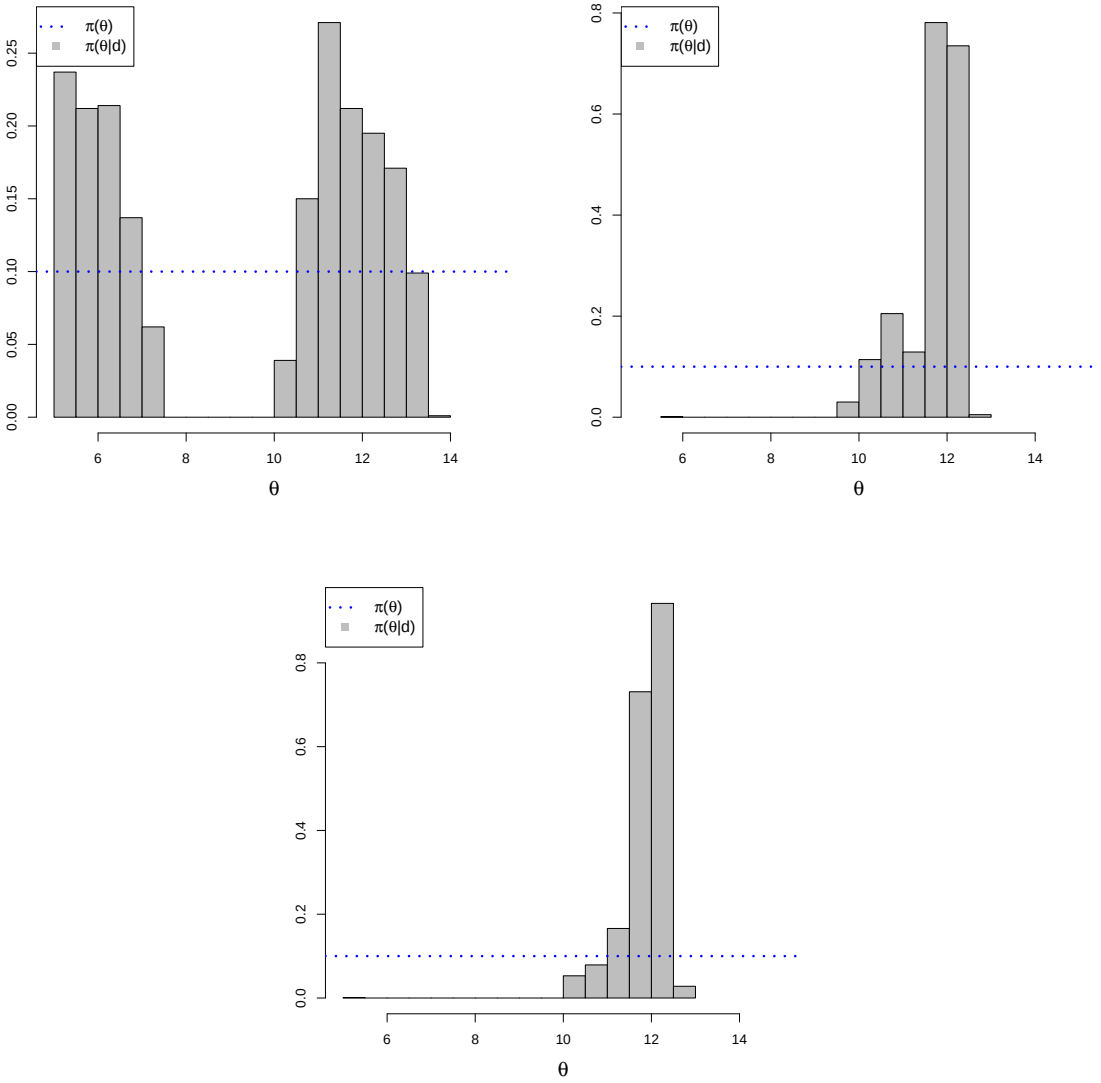
Remarque 2.4.2.1 *Pour l'exemple académique que nous avons traité, la librairie SAVE s'est montrée instable au choix de $\boldsymbol{\theta}^\star$ pour le calcul de $(\hat{\boldsymbol{\beta}}_b, \hat{\boldsymbol{\Psi}}_b)$ à partir de la vraisemblance des erreurs virtuelles*

$$\mathbf{z}^f - y_{\boldsymbol{\theta}^\star}(\mathbf{X}^f) \sim \mathcal{N}(h_b(\mathbf{X}^f)\boldsymbol{\beta}_b, \sigma_b^2 \Sigma_{\boldsymbol{\Psi}_b} + \lambda^2 \mathrm{I}_n). \tag{2.74}$$

Nous avons constaté en effet que pour obtenir de bonnes prédictions de $r(\mathbf{x})$, $\boldsymbol{\theta}^\star$ doit être choisi assez proche d'une valeur de $\boldsymbol{\theta}$ associée à une erreur suffisamment monotone (ici

$\boldsymbol{\theta} = 12$*). De plus, nous avons observé une incohérence entre la présentation de la documentation qui suppose* $\boldsymbol{\beta}_b = 0$ *et la procédure d'estimation implémentée dans SAVE qui ne tient pas compte de cette hypothèse et estime* $\boldsymbol{\beta}_b$*. Nous avons aussi remarqué que l'estimation conjointe de la variance du bruit* λ^2 *et de la variance* σ_b^2 *en maximisant la vraisemblance (2.74) aboutissait régulièrement à des valeurs aberrantes. Étant donné que la loi a posteriori de* λ^2 *est ensuite échantillonnée à partir de sa conditionnelle complète (2.67), la valeur estimée est moins utile ensuite, contrairement à la valeur des paramètres de corrélations* $\boldsymbol{\Psi}_b$ *qui doit être bien estimée. Nous avons donc modifié la procédure d'estimation en considérant* λ^2 *fixée à une valeur acceptable dans la procédure de maximisation de la vraisemblance (2.74).*

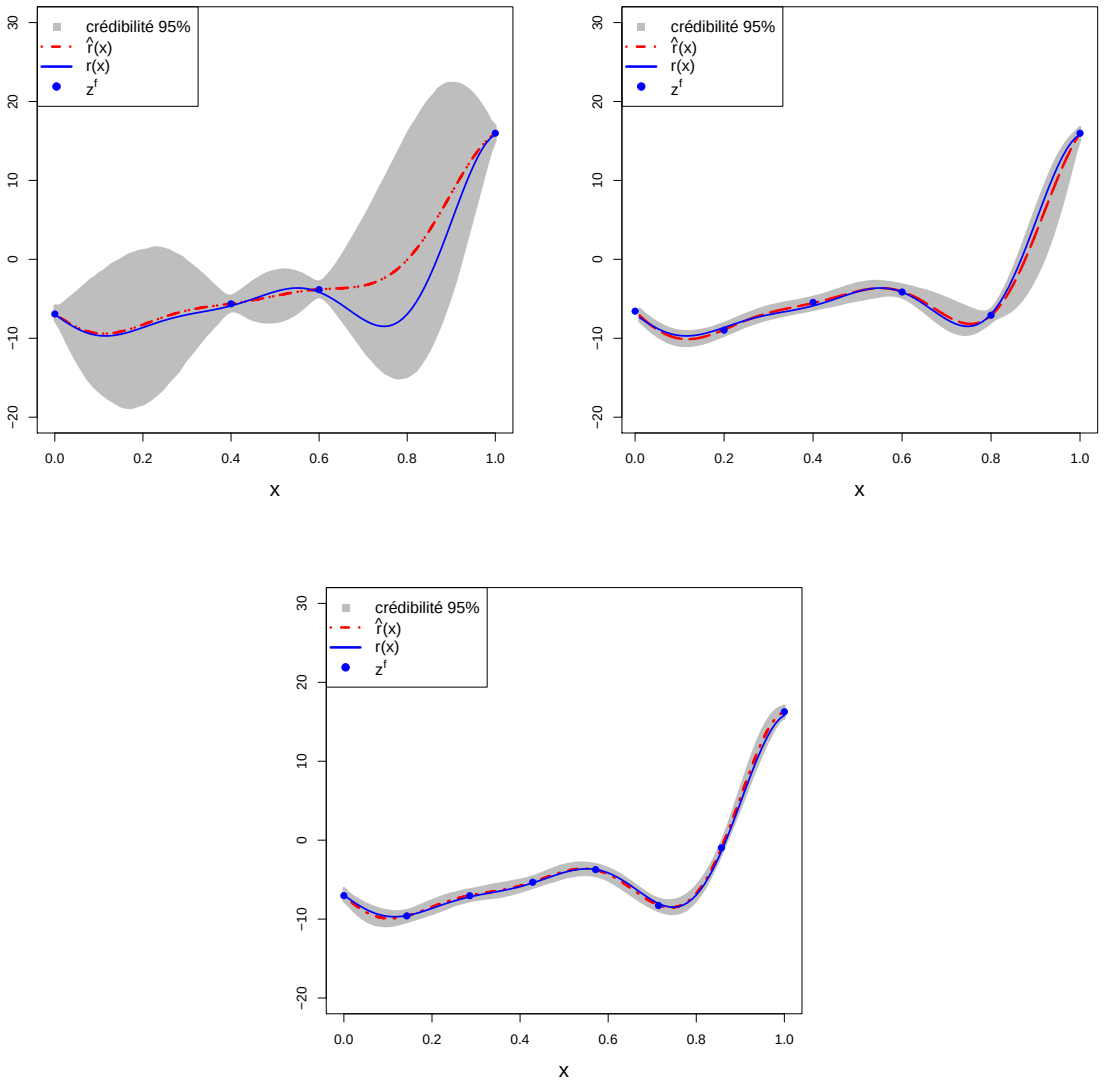
FIGURE 2.18 – *Méthode de Bayarri. À gauche : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$*. À droite : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$*. En bas : loi a posteriori de* θ *lorsque* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$.



Bayarri et al. (2007a) étendent le modèle (2.65) pour le calage et la validation des codes à réponse fonctionnelle qui modélisent des systèmes physiques qui évoluent en fonction du temps t :

$$r(\mathbf{x}, t) = y_{\boldsymbol{\theta}}(\mathbf{x}, t) + b(\mathbf{x}, t) \tag{2.75}$$

FIGURE 2.19 – *Méthode de Bayarri. À gauche : prédiction de* $r(\mathbf{x})$ *(2.72) sur* $[0,1]$ *calculée lorsque* $\mathbf{X}^f = (0, 0.4, 0.6, 1)^{\mathrm{T}}$. *À droite : prédiction calculée lorsque* $\mathbf{X}^f = (0, 0.2, 0.4, 0.6, 0.8, 1)^{\mathrm{T}}$. *En bas : prédiction calculée lorsque* $\mathbf{X}^f = (0, 0.14, 0.29, 0.43, 0.57, 0.71, 0.86, 1)^{\mathrm{T}}$. *Les processus gaussien sur le code et l'erreur à sont à moyenne constante. Le processus gaussien sur le code est construit à partir d'un plan d'expériences en grille de taille* $\mathrm{M} = 90$ *sur* $[0,1] \times [5,15]$.



La méthode est fondée sur une décomposition en base d'ondelettes de la réponse du code $y_{\boldsymbol{\theta}}(\mathbf{x}, t) = \sum_i w_i(\mathbf{x}, \boldsymbol{\theta})\psi_i(t)$ dans laquelle les coefficients $w_i(\mathbf{x}, \boldsymbol{\theta})$ sont « émulés » toujours à l'aide d'un processus gaussien.

2.4.3 Méthode de Higdon

Dans Higdon et al. (2004), le calage du code est effectué en supposant d'abord que $b(\mathbf{x}) = 0$, puis en introduisant $b(\mathbf{x})$ lorsque l'écart entre les réponses du code $y_{\boldsymbol{\theta}_i}(\mathbf{x})$ et les mesures $\mathbf{z}^f$ semble plus important qu'un bruit blanc et cela pour un nombre significatif de valeurs probables *a priori* $\boldsymbol{\theta}_i \in \mathscr{T}$. Cette progression ne figure pas dans les travaux de Kennedy et O'Hagan (2001a) et Bayarri et al. (2007b) où $b(\mathbf{x})$ est introduite d'emblée. Malheureusement, la prise en compte ou non de cette erreur dans Higdon et al. (2004) ne repose pas sur un critère quantitatif, ce que nous proposons dans le chapitre 3 dans le cas des codes linéaires en $\boldsymbol{\theta}$. Des tests numériques de la méthode ont déjà été effectués (http://www.gdr-mascotnum.fr/media/calibration_barbillon.pdf).

Signalons pour finir que Higdon et al. (2008) étendent le modèle (2.65) pour le calage et la validation des codes de calcul en grande dimension en considérant une décomposition en composantes principales de la réponse.

2.4.4 Comparaison des méthodes et interprétation des résultats

Hypothèses de modélisation Deux caractéristiques rendent les approches de Higdon et al. (2004) et Bayarri et al. (2007b) plus cohérentes que celle de Kennedy et O'Hagan (2001a) pour le calage et la validation d'un code de calcul :

1. le paramètre ρ est fixé à 1,
2. l'erreur de code est modélisé par un processus gaussien de moyenne constante $\boldsymbol{\beta}_b$ et les auteurs recommandent de poser $\boldsymbol{\beta}_b = 0$.

Ces deux hypothèses restrictives impliquent, d'une part que les réponses du code de calcul fournissent des prédictions à l'échelle de $r(\mathbf{x})$, et d'autre part que le code de calcul peut prédire de façon non biaisée la moyenne du système physique sur $\mathscr{X}$. On aimerait que ces deux hypothèses soient vérifiées en amont d'une procédure de calage et de validation. En effet, est-il bien raisonnable de calculer des prédictions d'un système physique à l'aide d'un code de calcul dont les réponses sont à une échelle différente et/ou translatées avec la réalité ? Nous croyons que si de tels problèmes subsistent, ils sont du ressort de la conception du code de calcul. Ensuite, si $\boldsymbol{\beta}_b = 0$ et qu'il existe une valeur $\boldsymbol{\tau} = \boldsymbol{\theta}^\star$ telle que $b_{\boldsymbol{\theta}^\star}(\mathbf{x}) = 0$ pour tout $\mathbf{x}$, alors le maximum de vraisemblance $\hat{\mathbf{t}}$ du modèle (2.65) converge vers $\boldsymbol{\theta}^\star$ sous l'hypothèse que le code n'est pas coûteux (Loeppky et al., 2006). Autrement dit, l'estimation de $\boldsymbol{\theta}$ dans le modèle (2.65) lorsque n est suffisamment grand coïncide avec l'estimation de $\boldsymbol{\theta}$ dans (2.20) sous l'hypothèse que le code n'est pas remplacé par un émulateur processus gaussien.

Techniques d'estimation utilisées La méthode de Higdon et al. (2004) propose d'estimer conjointement $\boldsymbol{\theta}$ et les hyperparamètres des processus gaussiens mis en jeux, tandis que les techniques proposées dans Kennedy et O'Hagan (2001a) et Bayarri et al. (2007b) ont en commun l'estimation des hyperparamètres du code à partir de la vraisemblance partielle des simulations $y(\mathbf{D}_{\mathrm{M}})$, appelée approche modulaire dans Liu et al. (2009). Cette démarche est assez intuitive dans le sens où seules les simulations $y(\mathbf{D}_{\mathrm{M}})$ influent sur l'estimation des hyperparamètres $(\boldsymbol{\Psi}_y, \sigma_y^2)$ du processus gaussien modélisant le code. En ce

qui concerne l'estimation des hyperparamètres $(\boldsymbol{\Psi}_b, \sigma_b^2, \lambda^2)$, Kennedy et O'Hagan (2001a) maximisent la vraisemblance des mesures physiques $\mathbf{z}^f$ conditionnellement aux simulations après avoir intégré par rapport à la loi *a priori* de $\boldsymbol{\theta}$ (2.53). Bayarri et al. (2007b) optent pour une approche plus hybride dans laquelle $\boldsymbol{\Psi}_b$ est estimé en maximisant la vraisemblance des erreurs virtuelles (2.74) calculée à partir d'une valeur *a priori* $\boldsymbol{\theta}^\star$, puis les lois *a posteriori* de σ_b^2 et λ^2 sont échantillonnées avec celles de $\boldsymbol{\theta}$ et des données latentes $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$ et $b(\mathbf{X}^f)$ dans un algorithme de Gibbs basé sur les conditionnelles complètes (2.66) à (2.69). Cela permet la construction naturelle d'un prédicteur de l'erreur de code (2.71), ce qui ne figure pas dans la méthode de KOH. Notons enfin que chaque nouvelle évaluation de (2.48) demande d'inverser une matrice de variance-covariance de taille $(n \times \mathrm{M}) \times (n \times \mathrm{M})$, ce qui peut être préjudiciable lorsque la taille du plan d'expériences M est importante. Combiné avec un paramètre $\boldsymbol{\theta}$ de grande dimension, l'échantillonnage de (2.48) peut alors devenir très coûteux.

Prédiction du système physique Le calcul des prédicteurs (2.60) et (2.72) en de nouvelles configurations d'entrée $\mathbf{x} \in \mathscr{X}$ repose sur l'expression de la loi conditionnelle $r(\mathbf{x})|\mathbf{d}$. Si de nouvelles simulations $y(\mathbf{D}_{new})$ étaient effectués, ces prédicteurs devraient être recalculés à partir de

$$\int_{\mathbf{u},\boldsymbol{\theta}} \pi(y_{\boldsymbol{\theta}}(\mathbf{x}), b(\mathbf{x})|\mathbf{d}, y(\mathbf{D}_{new}), \mathbf{u}, \boldsymbol{\theta}) \pi(\mathbf{u}, \boldsymbol{\theta}|\mathbf{d}, y(\mathbf{D}_{new})) \mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta}, \tag{2.76}$$

où $\pi(\mathbf{u}, \boldsymbol{\theta}|\mathbf{d}, y(\mathbf{D}_{new}))$ quantifie l'incertitude affectant $\boldsymbol{\theta}$ mise à jour à partir de $y(\mathbf{D}_{new})$ et dont le calcul nécessite de reconduire l'algorithme d'estimation de Gibbs décrit dans la section 2.4.2, ce qui n'est pas toujours possible pour des codes de calcul de grande dimension. C'est pourquoi, Bayarri et al. (2007b) conseillent de calculer à la place le prédicteur

$$\int_{\mathbf{u},\boldsymbol{\theta}} \pi(y_{\boldsymbol{\theta}}(\mathbf{x}), b(\mathbf{x})|\mathbf{d}, y(\mathbf{D}_{new}), \mathbf{u}, \boldsymbol{\theta}) \pi(\mathbf{u}, \boldsymbol{\theta}|\mathbf{d}) \mathrm{d}\mathbf{u}\mathrm{d}\boldsymbol{\theta}, \tag{2.77}$$

qui approxime (2.76) lorsque le nombre de nouvelles simulations $y(\mathbf{D}_{new})$ est négligeable par rapport à M.

Le fait que les prédicteurs (2.60) et (2.72) ne requièrent pas l'exécution du code peut être difficile à concevoir d'un point de vue opérationnel. Une solution alternative au calcul de (2.76) ou (2.77) lorsque des simulations supplémentaires $y(\mathbf{D}_{new})$ peuvent être effectuées, pourrait être de considérer la réponse du code $y_{\hat{\boldsymbol{\theta}}}(\mathbf{x})$ où $\hat{\boldsymbol{\theta}}$ est le mode ou la moyenne *a posteriori* de $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$, puis de calculer l'erreur de code à partir des erreurs $\mathbf{z}^f - y(\mathbf{D}_{\hat{\boldsymbol{\theta}}})$. Ce faisant, on se ramène au cas étudié dans la section 2.2.2.2.

Interprétation de $\pi(\boldsymbol{\theta}|\mathbf{d})$ La loi *a posteriori* $\pi(\boldsymbol{\theta}|\mathbf{d})$ quantifie l'incertitude *a posteriori* affectant $\boldsymbol{\theta}$ relativement à $\pi(\boldsymbol{\theta})$ et aux hypothèses *a priori* de processus gaussiens sur le code et l'erreur de code. Lorsque le paramètre $\boldsymbol{\theta}$ est un paramètre qui revêt une interprétation physique, $\pi(\boldsymbol{\theta}|\mathbf{d})$ peut-elle traduire l'incertitude sur la vraie valeur physique du paramètre $\boldsymbol{\theta}_{reel}$? Pour répondre à cette question, considérons un estimateur ponctuel $\boldsymbol{\theta}^\star$ de $\pi(\boldsymbol{\theta}|\mathbf{d})$, typiquement la moyenne ou le mode a *posteriori* de $\pi(\boldsymbol{\theta}|\mathbf{d})$. Loeppky et al. (2006) font apparaître explicitement la dépendance du système physique à $\boldsymbol{\theta}_{reel}$ en notant $r_{\boldsymbol{\theta}_{reel}}(\mathbf{x})$ au lieu de $r(\mathbf{x})$. On obtient,

$$\begin{aligned} b_{\boldsymbol{\theta}^\star}(\mathbf{x}) &= r_{\boldsymbol{\theta}_{reel}}(\mathbf{x}) - y_{\boldsymbol{\theta}^\star}(\mathbf{x}) \\ &= \big(r_{\boldsymbol{\theta}_{reel}}(\mathbf{x}) - y_{\boldsymbol{\theta}_{reel}}(\mathbf{x})\big) + \big(y_{\boldsymbol{\theta}_{reel}}(\mathbf{x}) - y_{\boldsymbol{\theta}^\star}(\mathbf{x})\big). \end{aligned} \tag{2.78}$$

Par suite, l'erreur de code estimée est égale à la somme de l'erreur de code vraie et de l'erreur de calage du code. L'équation (2.78) traduit encore le problème d'identifiabilité du modèle avec erreur de code, nous empêchant de relier la valeur estimée $\boldsymbol{\theta}^\star$ avec la vraie valeur physique $\boldsymbol{\theta}_{reel}$ sous-jacente au système physique. Dans le cas où $\boldsymbol{\theta}^\star$ est telle que $b_{\boldsymbol{\theta}^\star}(\mathbf{x}) = 0$, on obtient l'égalité

$$r_{\boldsymbol{\theta}_{reel}}(\mathbf{x}) = y_{\boldsymbol{\theta}^\star}(\mathbf{x}) \tag{2.79}$$

mais qui n'implique pas forcément $\boldsymbol{\theta}^\star = \boldsymbol{\theta}_{reel}$! Pour conclure, $\boldsymbol{\theta}$ doit être interprété comme le paramètre permettant le meilleur ajustement de la réponse du code aux mesures physiques étant donnée la structure des processus gaussiens sur le code et l'erreur de code.

Extension à la multi-fidélité Comme dans la section 2.2.2.2, il est possible d'étendre cette méthodologie de calage/validation lorsque plusieurs versions de code entachées d'une incertitude paramétrique reproduisent avec différents niveaux de précision le système physique d'intérêt $r(\mathbf{x})$. Goh et al. (2013) traitent le cas de deux niveaux de code :

$$\begin{cases} z_i^f = y^1_{\boldsymbol{\theta},\boldsymbol{\theta}_1}(\mathbf{x}_i^f) + b^1(\mathbf{x}_i^f) + \epsilon_i \\ y^1_{\boldsymbol{\theta},\boldsymbol{\theta}_1}(\mathbf{x}_i^f) = y^2_{\boldsymbol{\theta},\boldsymbol{\theta}_2}(\mathbf{x}_i^f) + b^2_{\boldsymbol{\theta},\boldsymbol{\theta}_1}(\mathbf{x}_i^f) \end{cases} \tag{2.80}$$

où $\boldsymbol{\theta}_1$ et $\boldsymbol{\theta}_2$ représentent respectivement des paramètres propres à chacun de ces deux niveaux de code et où $\boldsymbol{\theta}$ est un paramètre commun aux deux niveaux.

2.4.5 Modélisation alternative de l'erreur de code

Bachoc et al. (2014) modélisent l'erreur de code par un processus gaussien centré qu'ils interprètent comme une structure d'erreur corrélée (et non plus comme un émulateur *a priori* dans un contexte bayésien) dans un modèle de régression des mesures physiques sur la réponse du code paramétré par $\boldsymbol{\theta}$:

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i, \tag{2.81}$$

avec

$$b(.) \sim \mathcal{PG}\left(0, \sigma_b^2 \mathrm{K}_{\boldsymbol{\Psi}_b}(.)\right).$$

On parle alors de régression par processus gaussien plutôt que d'émulation. Notons qu'en terme de vocabulaire, σ_b^2 et $\boldsymbol{\Psi}_b$ deviennent des paramètres du modèle (et non plus des hyperparamètres d'un processus gaussien *a priori*). Cette approche permet d'éviter les problèmes d'identifiabilité soulevées dans la section 2.4. Bachoc et al. (2014) recommandent d'utiliser l'équation (2.81) lorsque l'amplitude des écarts entre les réponses du code et les mesures physiques $|z_i^f - y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)|$ ne correspond pas à l'amplitude déterminée par la variance λ^2 du bruit. Dans le cas où le code est linéaire, nous proposons dans le chapitre 3 une procédure statistique permettant de tester la présence de $b(.)$ lorsque λ^2 est inconnue.

2.4.6 Cas d'un code linéaire

Lorsque le code est linéaire, l'équation (2.81) se réécrit :

$$z_i^f = h(\mathbf{x}_i^f)^{\mathrm{T}}\boldsymbol{\theta} + b(\mathbf{x}_i^f) + \epsilon_i, \tag{2.82}$$

avec

$$b(.) \sim \mathscr{PG}\Big(0, \sigma_b^2 \mathrm{K}_{\boldsymbol{\Psi}_b}(.)\Big). \tag{2.83}$$

Par suite,

$$z(.) \sim \mathscr{PG}\Big(h(.)^{\mathrm{T}}\boldsymbol{\theta}, \sigma_b^2 \mathrm{K}_{\boldsymbol{\Psi}_b}(.) + \lambda^2 \mathrm{I}_n\Big). \tag{2.84}$$

L'équation (2.84) correspond à un modèle de processus gaussien « bruité », ce qui permet de simplifier à la fois l'estimation de $\boldsymbol{\theta}$ et la prédiction de $r(\mathbf{x})$. Bachoc et al. (2014) conduisent une estimation bayésienne de $\boldsymbol{\theta}$ tandis que λ^2 et les paramètres σ_b^2 et $\boldsymbol{\Psi}_b$ sont estimés par maximum de vraisemblance. Pour des codes non linéaires, les auteurs conseillent de se ramener à l'équation (2.82) en effectuant une linéarisation au voisinage d'une valeur *a priori* $\boldsymbol{\theta}^\star \in \mathscr{T}$. Cette opération est bien justifiée si $\boldsymbol{\theta}^\star \in \mathscr{T}$ constitue une bonne approximation de $\boldsymbol{\theta}$, ou alors si le domaine $\mathscr{T}$ est suffisamment petit pour pouvoir linéariser le code en n'importe quelle valeur $\boldsymbol{\theta}^\star \in \mathscr{T}$. Dans ce dernier cas, la version linéarisée du code sur $\mathscr{T}$ peut être considérée comme une approximation suffisamment bonne du code pour tout $\boldsymbol{\tau} \in \mathscr{T}$.

Nous terminons cet état de l'art par la présentation de deux méthodes adaptées au calage et à la validation des codes de calcul lorsque le paramètre $\boldsymbol{\theta}$ est de grande dimension et/ou lorsque les entrées et/ou la réponse du code appartiennent à des espaces de grande dimension. Dans ces situations, les méthodes d'échantillonnage par MCMC présentées dans la section 2.4 sont très coûteuses à mettre en œuvre, voire inadaptées.

2.5 Méthodes de calage et de validation pour la grande dimension

2.5.1 L'analyse bayésienne linéaire

Une famille de méthodes fondée sur la théorie de la statistique bayésienne linéaire dont fait l'objet l'ouvrage de Goldstein et Wooff (2007) peut remplacer l'inférence bayésienne classique lorsque la dimension des quantités mises en jeu est importante (voir annexe B).

Nous nous basons sur l'article de Craig et al. (2001) qui traite un problème de calage de paramètres en ingénierie réservoir avec pour objectif la prédiction de données de pression dans des puits de production de gaz. Contrairement à la section précédente, le système physique est étudié en fonction du temps t à la place du vecteur de variables $\mathbf{x}$. La réponse du code de calcul se compose des mesures de pression sur une période de temps que l'on peut découper en une période passée p et une période à venir a. Elle peut donc s'écrire sous la forme $y_{\boldsymbol{\theta}}(.) = \big(y_{\boldsymbol{\theta}}(p), y_{\boldsymbol{\theta}}(a)\big) \in \mathbb{R}^m$. Comme dans les méthodes étudiées dans la section 2.4, une fonction d'erreur de code relie les réponses du code de calcul au système physique d'intérêt, ce qui donne pour la période passée

$$r(p) = y_{\boldsymbol{\theta}}(p) + b(p)$$

avec $b(p) \sim \mathcal{N}(0, \Sigma_p)$ et pour la période à venir

$$r(a) = y_{\boldsymbol{\theta}}(a) + b(a)$$

avec $b(a) \sim \mathcal{N}(0, \Sigma_a)$. Le code de calcul est coûteux, ce qui implique que la base d'apprentissage des réponses du code $y(\mathbf{D}_{\mathrm{M}}) := \{y_{\boldsymbol{\tau}^1}(.), \cdots, y_{\boldsymbol{\tau}^{\mathrm{M}}}(.)\}$ est de taille réduite. Pour répondre au problème de calage et à la validation du code, il s'agit comme dans la section

2.4 de calculer la loi *a posteriori* de $\boldsymbol{\theta}$, notée $\pi(\boldsymbol{\theta}|y(\mathbf{D}_{\mathrm{M}}),\mathbf{z}^f(p))$, puis la loi de prédiction de $r(a)$ donnée par

$$\pi(r(a)|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}))=\int \pi(r(a)|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}))\mathrm{d}\boldsymbol{\theta}. \tag{2.85}$$

Le code étant coûteux, il est remplacé par un émulateur processus gaussien de moyenne

$$\mu(\boldsymbol{\theta})=(\mu_p(\boldsymbol{\theta}),\mu_a(\boldsymbol{\theta})),$$

et de matrice de variance-covariance

$$\mathrm{V}(\boldsymbol{\theta})=\begin{pmatrix}\mathrm{V}_p(\boldsymbol{\theta}),\mathrm{V}_{a,p}(\boldsymbol{\theta})\\ \mathrm{V}_{p,a}(\boldsymbol{\theta}),\mathrm{V}_a(\boldsymbol{\theta})\end{pmatrix}. \tag{2.86}$$

Notons $\Sigma_b(p,a)$ la structure de corrélation *a priori* entre $b(p)$ et $b(a)$ et définissons

$$\mathrm{C}(\boldsymbol{\theta})=\begin{pmatrix}\mathrm{C}_p(\boldsymbol{\theta}),\mathrm{C}_{a,p}(\boldsymbol{\theta})\\ \mathrm{C}_{p,a}(\boldsymbol{\theta}),\mathrm{C}_a(\boldsymbol{\theta})\end{pmatrix} \tag{2.87}$$

avec $\mathrm{C}_p(\boldsymbol{\theta})=\mathrm{V}_p(\boldsymbol{\theta})+\Sigma_b(p)+\lambda^2\mathrm{I}$, $\mathrm{C}_a(\boldsymbol{\theta})=\mathrm{V}_a(\boldsymbol{\theta})+\Sigma_b(a)$, $\mathrm{C}_{p,a}(\boldsymbol{\theta})=\mathrm{V}_{p,a}(\boldsymbol{\theta})+\Sigma_b(p,a)$ et $\mathrm{C}_{a,p}(\boldsymbol{\theta})=\mathrm{C}_{p,a}(\boldsymbol{\theta})^{\mathrm{T}}$. En utilisant les formules du conditionnement gaussien, on peut montrer que $\pi(r(a)|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\theta})$ suit une loi gaussienne de moyenne

$$m_{a|p}(\boldsymbol{\theta})=\mu_a(\boldsymbol{\theta})+\mathrm{C}_{p,a}(\boldsymbol{\theta})\mathrm{C}_p(\boldsymbol{\theta})^{-1}(\mathbf{z}^f(p)-\mu_p(\boldsymbol{\theta}), \tag{2.88}$$

et de variance

$$\Sigma a|p(\boldsymbol{\theta})=\mathrm{C}_a(\boldsymbol{\theta})-\mathrm{C}_{a,p}(\boldsymbol{\theta})\mathrm{C}_p(\boldsymbol{\theta})^{-1}\mathrm{C}_{p,a}(\boldsymbol{\theta}). \tag{2.89}$$

La loi *a posteriori* $\pi(\boldsymbol{\theta}|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}))$ n'a pas de forme explicite. Son échantillonnage ainsi que celui de $\pi(r(a)|\mathbf{z}^f(p),y(\mathbf{D}_{\mathrm{M}}),\boldsymbol{\theta})$ requièrent donc des méthodes MCMC comme celles mises en œuvre dans Kennedy et O'Hagan (2001a) ou Bayarri et al. (2007b). Pour des problèmes en grande dimension comme ceux abordés en ingénierie réservoir, elles sont inapplicables. Dans Craig et al. (2001), le paramètre $\boldsymbol{\theta}$ est de dimension 40 !

Une approche alternative consiste à remplacer l'approche bayésienne par une approche bayésienne linéaire qui nécessite des hypothèses *a priori* uniquement sur la moyenne et la variance des quantités incertaines. Les croyances *a priori* sur $(\mathbf{z}^f(p),r(a))$ sont alors êtres mises à jour compte tenu des mesures $\mathbf{z}^f(p)$ via les deux équations suivantes :

$$\mathbb{E}_{\mathbf{z}^f(p)}[r(a)]=\mathbb{E}[r(a)]+Cov(r(a),\mathbf{z}^f(p))\mathbb{V}[\mathbf{z}^f(p)]^{-1}(\mathbf{z}^f(p)-\mathbb{E}[\mathbf{z}^f(p)]) \tag{2.90}$$

qui est la moyenne du système physique « ajustée » par les observations $\mathbf{z}^f(p)$ et

$$\mathbb{V}_{\mathbf{z}^f(p)}[r(a)]=\mathbb{V}[r(a)]+Cov(r(a),\mathbf{z}^f(p))\mathbb{V}[\mathbf{z}^f(p)]^{-1}Cov(\mathbf{z}^f(p),r(a)) \tag{2.91}$$

qui permet de quantifier l'incertitude de prédiction autour de $\mathbb{E}_{\mathbf{z}^f(p)}[r(a)]$. Les expressions de $\mathbb{E}_{\mathbf{z}^f(p)}[r(a)]$ et $\mathbb{V}_{\mathbf{z}^f(p)}[r(a)]$ sont obtenues en considérant les moyennes et variances ajustées du simulateur qui sont égales ici à $\mu(\boldsymbol{\theta})$ et $\mathrm{V}(\boldsymbol{\theta})$. Ensuite, les croyances sur $y_{\boldsymbol{\theta}}(.)$, notées μ et V sont calculées à partir d'une loi *a priori* $\pi(\boldsymbol{\theta})$. On obtient

$$\mu=\mathbb{E}_{\boldsymbol{\theta}}[\mu(\boldsymbol{\theta})]=\int \mu(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{2.92}$$

et en utilisant la formule de la covariance totale

$$\mathrm{V}=\mathbb{E}_{\boldsymbol{\theta}}[\mathrm{V}(\boldsymbol{\theta})]+\mathbb{V}_{\boldsymbol{\theta}}[\mu(\boldsymbol{\theta})]. \tag{2.93}$$

Par suite, l'ajustement linéaire de $r(a)$ par rapport aux mesures passées $\mathbf{z}^f(p)$ vaut :

$$m_{a|p} = \mu_a + C_{p,a}C_p^{-1}(\mathbf{z}^f(p) - \mu_p) \tag{2.94}$$

et

$$\Sigma_{a|p} = C_a - C_{a,p}C_p^{-1}C_{p,a} \tag{2.95}$$

avec $C_{p,a} = V_{p,a} + \Sigma_b(p,a)$, $C_a = V_a + \Sigma_b(a)$ et $C_p = V_p + \Sigma_b(p) + \lambda^2 I$. L'incertitude *a posteriori* n'est traduite ici qu'en terme des deux premiers moments sans hypothèse sous-jacente sur la forme de la distribution. Les équations (2.94) et (2.95) quantifient l'incertitude sur les prévisions futures.

Des diagnostics de comparaison des croyances *a priori* avec les mesures de pression permettent de juger du bien fondé de la méthode. Par exemple, si $D_p := \mathbf{z}^f(p) - \mu_p$ est trop éloigné de 0 par rapport à la variance de $\mathbf{z}^f(p)$, alors les hypothèses *a priori* devraient être remises en cause.

La principale différence entre l'approche bayésienne linéaire et l'approche bayésienne classique réside dans le traitement de l'incertitude paramétrique. Dans l'approche bayésienne linéaire, l'incertitude du système physique est calculée à partir des données disponibles et des croyances *a priori* sur $\boldsymbol{\theta}$, sans étape de calage (sans calcul de la loi *a posteriori* de $\boldsymbol{\theta}$). Cette manière de procéder peut mener à de larges plages d'incertitude notamment si l'on dispose de peu d'expertise pour construire $\pi(\boldsymbol{\theta})$ et si l'émulateur processus gaussien sur $\mathcal{T}$ est peu précis. Goldstein et Rougier (2006) remédient à ce problème en proposant une approche bayésienne linéaire plus sophistiquée qui combine calage et prédiction. Une approche alternative consiste à conduire de façon alternée une étape de calage avec une étape d'analyse bayésienne linéaire. L'étape de calage [10] s'appuie alors sur une *mesure d'implausibilité* visant à identifier les valeurs de $\boldsymbol{\theta}$ incompatibles *a priori* avec les mesures physiques (Andrianakis et al., 2015).

2.5.2 Les méthodes d'assimilation de données

En grande dimension, les méthodes dites d'assimilation de données pourraient être appliquées au calage et à la validation des codes de calcul coûteux. Dans le domaine des prévisions météorologique et de l'ingénierie pétrolière, ces méthodes sont souvent utilisées avec succès pour mettre à jour l'état d'un système caractérisé au temps t par le paramètre $\boldsymbol{\theta}_t$, à l'aide des mesures physiques observées au cours du temps et d'un code de calcul de prévision du système. L'assimilation de données s'appuie sur l'estimateur du *filtre de Kalman* qui se compose de deux étapes successives : la prédiction du système à partir de l'état courant $\boldsymbol{\theta}_{t-1}$ et la mise à jour de l'incertitude paramétrique $\boldsymbol{\theta}_{t-1} \rightarrow \boldsymbol{\theta}_t$ à partir des nouvelles mesures observées au temps t. L'article de Higdon et al. (2013) constitue une référence instructive dans laquelle les auteurs comparent la formulation bayésienne de calage et de validation étudiée tout au long de ce chapitre, avec l'approche du filtre de Kalman d'ensemble (EnKF). En particulier, les auteurs remarquent que l'EnKF s'appuie sur une modélisation linéaire du code en fonction de $\boldsymbol{\theta}$ contrairement aux approches bayésienne que nous avons présentées qui s'appuient sur un émulateur processus gaussien pouvant capturer de possibles interactions entre les composantes de $\boldsymbol{\theta}$. Pour plus de détails à propos des méthodes d'assimilation de données, nous renvoyons à l'ouvrage de Evensen (2009).

10. couramment appelée *History Matching*

2.6 Conclusion

Dans ce chapitre, nous avons produit un état de l'art des méthodes statistiques pour le calage et la validation des codes de calcul. Elles consistent à quantifier l'incertitude de prédiction d'un système physique d'intérêt en utilisant à la fois les mesures physiques disponibles et les réponses du code. Nous avons largement approfondi les méthodes mises en œuvre lorsque le code est à la fois coûteux, et entaché d'une incertitude affectant ses paramètres. Dans cette situation, les méthodes statistiques qui utilisent l'émulation du code à l'aide d'un processus aléatoire gaussien permettent d'abord de mettre à jour cette incertitude paramétrique (on parle alors de calage du code), puis de calculer l'incertitude de prédiction du système physique pour répondre à l'objectif de la validation. Le prédicteur est calculé, soit seulement à partir des réponses du code de calcul, soit à partir des réponses du code corrigées par l'erreur de code estimée. La formulation statistique qui prend en compte une telle erreur induit des problèmes d'identifiabilité largement discutés dans la littérature et que nous avons illustrés sur un exemple académique.

Plusieurs pistes de travail émergent de cette étude bibliographique. D'une part, l'introduction de l'erreur de code devrait être mieux justifiée ; et pour cela, nous proposons dans le chapitre 3 une procédure bayésienne de sélection de modèle (Berger et Pericchi, 1996). D'autre part, la précision du calage de code lorsque les simulations sont coûteuses s'est avérée très sensible au choix du plan d'expériences. Pour améliorer la robustesse du calage vis à vis du choix de celui-ci, nous proposons au chapitre 4 de nouveaux algorithmes séquentiels fondés sur le critère de l'amélioration espérée (Jones et al., 1998).

Bibliographie

AIAA (1998). Guide for the verification and validation of computational fluid dynamics simulations. *Reston VA : American Institute of Aeronautics and Astronautics*, AIAA-G-077-1998. 15

Andrianakis, I., Vernon, I., McCreesh, N., McKinley, T., Oakley, J., Nsubuga, R., Goldstein, M., et White, R. (2015). Bayesian history matching of complex infectious disease models using emulation : A tutorial and a case study on hiv in uganda. *PLOS Computaionnal Biology*, 11(1). 49

ASME (2009). *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*. American Society of Mechanical Engineers. 15

Bachoc, F., Blois, G., Garnier, J., et Martinez, J. (2014). Calibration and improved prediction of computer models by universal kriging. *Nuclear Science and Engineering*, 176(1) :81–97. 46, 47

Barbillon, P., Celleux, G., Grimaud, A., Lefebvre, Y., et Rocquigny (De), E. (2011). Non linear methods for inverse statistical problems. *Computational Statistics and Data Analysis*, 55(1) :132–142. 12

Bayarri, M. J., Berger, J., Cafeo, J., Garcia-Donato, G., Liu, F., Palomo, J., Parthasarathy, R., Paulo, R., Sacks, J., et Walsh, D. (2007a). Computer model validation with functional output. *The Annals of Statistics*, 35(5) :1874–1906. 42

Bayarri, M. J., Berger, J. O., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., et Tu, J. (2007b). A framework for validation of computer models. *Technometrics*, 49(2) :138–154. 15, 31, 38, 41, 44, 45, 48

Berger, J. et Pericchi, L. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433) :109–122. 50

Bontemps, S., Kaemmerlen, A., Le-Berre, R., et Mora, L. (2013). La fiabilité d'outils de simulation thermique dynamique dans le contexte des bâtiments basse consommation. *Submission SFT*. 14

Brynjarsdottir, J. et O'Hagan, A. (2014). Learning about physical parameters : The importance of model discrepancy. *Inverse Problems*, 30 :114007. 38

Campbell, K. (2006). Statistical calibrations of computer simulations. *Reliability Engineering and System Safety*, 91(10-11) :1358–1363. 22

Cover, T. et Thomas, J. (1991). *Elements of Information Theory*. Wiley-Interscience. 26

Cox, D., Park, J., et Clifford, E. (2001). A statistical method for tuning a computer code to a data base. *Computational Statistics and Data Analysis*, 37(1) :77–92. 22, 24

Craig, P., Goldstein, M., Rougier, J., et Seheult, A. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, 96(454). 47, 48

Currin, C., Mitchell, T., Morris, M., et Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the Statistical Association*, 86(416) :953–963. 18

Denis, J. et Scutari, M. (2014). *Réseaux bayésiens avec R*. EDP Sciences-Pratique R. 22

Evensen, G. (2009). *Data assimilation : the ensemble Kalman filter*. Springer-Verlag. 49

Ferson, S. et Ginzburg, L. (1996). Different methods are needed to propagate ignorance and variability. *Reliability Engineering and System Safety*, 54(2-3) :133–144. 13

Ferson, S., Kreinovich, V., Ginzburg, D., Myers, D., et Sentz, K. (2003). Constructing probability boxes and dempster-shafer structures. Technical report, SAND2004-3072, Albuquerque, Sandia National Laboratories. 13

Forrester, A., Sobester, A., et Keane, A. (2007). Multi-fidelity optimization via surrogate modelling. *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, 463 :3251–3269. 19

Goh, J., Bingham, D., Holloway, J., Grosskopf, M., Kuranz, C., et Rutter, E. (2013). Prediction and computer model calibration using outputs from multi-fidelity simulators. *Journal of the American Statistical Association*, 55(4) :501–512. 46

Goldstein, D. et Wooff, D. (2007). *Bayes Linear Statistics, Theory and Methods*. Wiley. 47

Goldstein, M. et Rougier, J. (2006). Bayes linear calibrated prediction for complex systems. *Journal of the American Statistical Association*, 101(475) :1132–1143. 49

Hankin, R. (2005). Introducing BACCO, an R bundle for Bayesian analysis of computer output. *Journal of Statistical Software*, 14(16) :1–21. 35

Higdon, D., Gattiker, J., Lawrence, E., Jackson, C., Tobis, M., Pratola, M., Habib, S., K., H., et Price, S. (2013). Computer model calibration using the ensemble Kalman filter. *Technometrics*, 55(4) :488–500. 49

Higdon, D., Gattiker, J., Williams, B., et Rightley, M. (2008). Computer model calibration using high dimensional output. *Journal of the American Statistical Association*, 103(482) :570–583. 44

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J., et Ryne, R. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2) :448–466. 31, 44

Jones, D., Schonlau, M., et Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4) :455–492. 50

Kennedy, M. et O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1) :1–13. 18, 21

Kennedy, M. et O'Hagan, A. (2001a). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B, Methodological*, 63(3) :425–464. 12, 16, 31, 33, 38, 44, 45, 48

Kennedy, M. et O'Hagan, A. (2001b). Supplementary details on Bayesian calibration of computer models. *Internal Report*. 34

Kiureghian, A. et Ditlevsen, O. (2009). Aleatory or epistemic ? does it matter ? *Structural Safety*, 31(2) :105–112. 12

Le Gratiet, L. (2013). Bayesian analysis of hierarchical multi-fidelity codes. *SIAM/ASA Journal of Uncertainty Quantification*, 1(1) :244–269. 19, 20, 21

Liu, F., Bayarri, M., et Berger, J. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1) :119–150. 25, 44

Loeppky, D., Bingham, D., et Welch, W. (2006). Computer model calibration or tuning in practice. Technical report, Department of Statistics, University of British Columbia, Canada. 44, 45

Oberkampf, W. et Barone, M. (2006). Measures of agreement between computation and experiment : Validation metrics. *Journal of Computational Physics*, 217(1) :5–36. 15, 16

Phillips, T. et Roy, J. (2011). Evaluation of extrapolation-based discretization error and uncertainty estimators. In *49th AIAA Aerospace Sciences Meeting inluding the New Horizons Forum and Aerospace Exposition, 4-7 January, Orlando Floridia*. 14

Qian, P. Z. G. et Wu, C. J. (2008). Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2) :192–204. 21

Rebba, R. (2008). *Model valiation and design under uncertainty*. PhD thesis, Faculty of the Graduate School of Vanderbilt University. 15

Rebba, R., Huang, S., Liu, Y., et Mahadevan, S. (2006). Statistical validation of simulation models. *Int. J. Materials and Product Technology,*, 25(1-3). 16

Roache, P. (1998). Verification of codes and calculations. *AIAA Journal*, 36(5) :696–702. 14

Robert, C. (1996). *Méthodes de Monte Carlo par Chaines de Markov*. Economica. 23

Roy, C. et Oberkampf, W. (2011). A comprehensive framework for verification, validation and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28) :2131–2144. 13, 16

Sacks, J., W.J., W., Mitchell, T., et Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4) :409–423. 15, 19

Santner, T., Williams, B., et Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag. 10, 11

Trucano, T., Swiler, L., Igusa, T., Oberkampf, W., et Pilch, M. (2006). Calibration, validation and sensitivity analysis : What's what. *Reliability Engineering and System Safety*, 91(10-11) :1331–1357. 14

Wang, S., Chen, W., et Tsui, K. (2009). Bayesian validation of computer models. *Technometrics*, 51(4) :439–451. 19

Chapter 3

Bayesian model selection for the validation of computer codes

Contents

Abstract Complex physical systems are increasingly modeled by computer codes which are implemented as mathematical representations of the reality. Many recent frameworks devoted to the validation of these codes aim at predicting a physical quantity of interest by using a code prediction, either pure or discrepancy-corrected, as advocated by Bayarri et al. (2007). In the present paper, we introduce a testing procedure to properly assess whether or not a code discrepancy should be taken into account between the code output and the physical system of interest.

For computer codes which depend on uncertain parameters, this problem of model selection can be addressed in a Bayesian setting, requiring the specification of prior distributions which are known as having a great impact on the result. Furthermore, in the absence of expert opinion, some of these priors can be improper, which is a major barrier for computing the Bayes factor. We propose to address this issue by using the *intrinsic Bayes factor*, originally developed by Berger and Pericchi (1996) to replace the ill-defined Bayes factor when improper priors are used. For computer codes depending linearly with respect to the code parameters, the computation of the IBF is made easier thanks to some explicit marginalization. In addition, we address a special case where the IBF is equal to the standard Bayes factor when the right Haar prior is specified on the code parameters and the scale of the code discrepancy.

On simulated data, we put in light a confounding effect between the code discrepancy and the linear code, then we compare the results according to the choice of the prior distribution. Finally, the IBF is computed for an industrial computer code used for monitoring power plant production.

Keywords: Bayes factor, Bayesian calibration, Gaussian process regression

Résumé Les systèmes physiques complexes sont de plus en plus souvent modélisés par des codes de calculs construits à partir de modélisations mathématiques de la réalité. De nombreux travaux récents traitent de la validation des codes comme la démarche visant à calculer un prédicteur du système physique, soit à partir des réponses du code, soit à partir des réponses du code corrigées par une “erreur de code” suivant la méthodologie proposée par Bayarri et al. (2007). Dans cet article, nous présentons une procédure de test visant à détecter si une telle fonction d’erreur entre les réponses du code et le système physique d’intérêt doit être prise en compte ou non pour le calcul du prédicteur.

Pour les codes de calcul entachés d’une incertitude paramétrique, ce problème de sélection de modèle peut être abordé d’un point de vue bayésien nécessitant la construction de lois *a priori* pouvant influencer grandement les résultats de la procédure de test. En l’absence d’avis d’experts, certaines lois *a priori* peuvent être impropres, ne permettant pas de définir correctement le facteur de Bayes. Nous proposons pour contourner cet obstacle de calculer le *facteur de Bayes intrinsèque* introduit par Berger and Pericchi (1996), qui se substitue au facteur de Bayes habituel quand des lois *a priori* impropres sont utilisées. Pour les codes de calcul linéaires par rapport aux paramètres, le calcul du facteur de Bayes intrinsèque est facilité par l’expression analytique de certaines vraisemblances marginales. En outre, nous traitons un cas particulier pour lequel le facteur de Bayes intrinsèque est égal au facteur de Bayes habituel lorsque la loi *a priori* de Haar est utilisée sur les paramètres du code et sur le paramètre d’échelle de l’erreur de code.

En calculant l’IBF sur données simulées, nous mettons en évidence un effet de compensation entre le code et l’erreur de code, puis nous comparons les résultats obtenus en fonction du choix de la loi *a priori*. Nous terminons en calculant l’IBF pour un code de calcul industriel dédié à la surveillance d’une installation de production d’électricité.

Mot-clés: Facteur de Bayes, Calage bayésien, Régression par processus gaussien.

3.1 Introduction

Because of their complexity, physical systems are often modeled by computer codes, which are used as a proxy for understanding the reality when the field experiments are either impractical or economically expensive. Let $r(\mathbf{x}) \in \mathbb{R}$ be a physical quantity of interest with respect to an input vector $\mathbf{x} \in \mathscr{X} \subset \mathbb{R}^d$, which can consist of a vector of control variables $\mathbf{x}^c$ and a vector of random variables $\mathbf{X}^e$ (Santner et al., 2003). A computer code can be seen as a parametric function $y_{\boldsymbol{\tau}}(\mathbf{x})$ where $\boldsymbol{\tau}$ is a vector of parameters which has no observable counterpart in the reality. It is referred to as a calibration or tuning parameter whether it has a physical interpretation or not (Gang et al., 2009; Loeppky et al., 2006).

Generally speaking, code validation questions the capability of the computer code to provide an accurate prediction of $r(\mathbf{x})$. This is a crucial issue, particularly in the field of risk assessment, where computer codes are used to assess the probability of failure of real systems. Over the years, many attempts have been made to standardize the terminology of validation, however several disctinct definitions are still associated to this concept. For instance, a naive definition would associate validation to a binary response either yes or no to the question "Can the computer code represent adequately the reality?". The American Institute of Aeronautics and Astronautics (AIAA) stresses that validation should be context dependent (AIAA, 1998). Their definition is in favor of conducting the validation through a two-steps procedure which should consist of assessing the uncertainty affecting the predictions of $r(\mathbf{x})$ based on both the code outputs and the field measurements, then deciding whether or not the code is valid in view of its intended uses. This paper focuses on the first step.

The methods dedicated to the uncertainty quantification of physical systems thanks to numerical predictions are most often based on statistical treatments. Oberkampf and Barone (2006) introduce some validation metrics in order to provide confidence intervals of the discrepancy between the code outputs and the reality when the parameter $\boldsymbol{\theta}$ is known. Roy and Oberkampf (2011) consider a method where the epistemic uncertainty tainting $\boldsymbol{\theta}$ and the aleatory uncertainty tainting $\mathbf{X}^e$ are propagated to the code output in such a way they can be distinguished through a probability box (Ferson et al., 2003). Unfortunately, such methods become impractical when the code runs are time-expensive.

For computer codes having no aleatory uncertainty in the inputs ($\mathbf{X}^e = 0$), Bayarri et al. (2007) circumvent this obstacle by using a Bayesian framework where the code is replaced by a Gaussian process emulator which is cheap to evaluate (Sacks et al., 1989). Such an emulator is now a common way of specifying a prior structure on the code output in a Bayesian setting (Currin et al., 1991). In addition, the method takes explicitly into account a discrepancy function $b(\mathbf{x})$ between the code and the reality. This discrepancy function has been introduced by Kennedy and O'Hagan (2001) who have argued that any computer code should be considered as an imperfect representation of the reality. Hence, $b(\mathbf{x})$ is defined as a nonzero function, such that:

$$b(\mathbf{x}) = r(\mathbf{x}) - y_{\boldsymbol{\theta}}(\mathbf{x}) \tag{3.1}$$

where the dependency of b to $\boldsymbol{\theta}$ can be removed as $\boldsymbol{\theta}$ is the optimal value of the parameter.

Kennedy and O'Hagan (2001) and Bayarri et al. (2007) jointly estimate the discrepancy $b(\mathbf{x})$ and $\boldsymbol{\theta}$ thanks both to the available field measurements and a limited number of simulations. Due to this discrepancy function, the computer code output alone cannot provide accurate predictions of $r(\mathbf{x})$ and instead a predictor of $r(\mathbf{x})$ should be calculated as a discrepancy-corrected prediction.

Quite surprisingly however, no mathematical justification of the presence of $b(\mathbf{x})$ is provided, in either Kennedy and O'Hagan (2001) or Bayarri et al. (2007), even though this

aditional term makes the inference significantly more complicated, as well as the interpretation of the results, and could be confusing for numerical practitioners and engineers who have constructed the computer code. From their point of view, the code is expected to be close to the real system, or at least to be an unbiased representation of it.

We thus believe that a validation framework should include a statistical procedure to test whether or not the code is correct in the sense that the discrepancy $b(\mathbf{x})$ is significantly different from zero. Loeppky et al. (2006) propose a likelihood ratio test to assess if $b(\mathbf{x})$ is significant. However, such frequentist tests are usually strongly biased toward type I (false rejection) error rate control, which means that they allow to assess, with a given level of confidence, the existence of a bias $b(\mathbf{x})$, but not its absence. Instead, we advocated the use of Bayesian model selection, which allows us to assess both the presence and absence of a model discrepancy $b(\mathbf{x})$ with the same confidence level.

In Section 3.2, the validation procedure of Bayarri et al. (2007) is detailed. This methodology represents the starting point of our work. We will end this section by emphasizing that a statistical testing could enable a better understanding of the assumption (3.1) from an engineering perspective. In Section 3.3, the theory of Bayesian model selection is quickly recalled, then we pay attention on the *intrinsic Bayes factor*. In Section 3.4, calculations of the IBF are performed by assuming the code is linear with respect to $\boldsymbol{\theta}$. In Section 3.5, the IBF is applied to an academic example, then we look at the impact of several prior distributions on the results. We also put in light a confounding effect between the code and the code discrepancy. In Section 3.6, the IBF is applied to the validation of an industrial computer code used for monitoring power plant production. We will end in Section 3.7 by giving some perspectives for future works.

3.2 Statistical modeling for code validation

In this first section, we recall the main features of the validation procedure detailed in Bayarri et al. (2007). It can be applied to computer codes which aim at predicting a physical quantity of interest $r(\mathbf{x})$ where

— the input vector $\mathbf{x} = (x_1, \cdots, x_d)^{\mathrm{T}} \in \mathscr{X} \subset \mathbb{R}^d$ is controllable, that is $\mathbf{X}^e = 0$;

— the optimal value of the code parameter $\boldsymbol{\theta}$ is uncertain and needs to be estimated. This stage is called calibration.

Bayarri et al. (2007) assume a discrepancy function $b(\mathbf{x})$ between the code output and the physical system which was introduced in the seminal work of Kennedy and O'Hagan (2001) devoted to code calibration. Bayarri's procedure exploits this idea in the field of code validation, when a calibration stage is required. After estimating $\boldsymbol{\theta}$ in a Bayesian setting, Bayarri et al. (2007) pay attention to the construction of tolerance bounds on $r(\mathbf{x})$. Let us now give more details about the statistical framework.

Let $\mathbf{z}^f = (z_1^f, \cdots, z_n^f)^{\mathrm{T}}$ be the vector of available field measurements of size n over the input physical design $\mathbf{X}^f = [\mathbf{x}_1^f, \cdots, \mathbf{x}_n^f]^{\mathrm{T}} \in \mathrm{M}_{n,d}(\mathbb{R})$. For $1 \leq i \leq n$:

$$z_i^f = r(\mathbf{x}_i^f) + \epsilon_i \tag{3.2}$$

where

$$\epsilon_i \sim \mathscr{E}_i \underset{i.i.d.}{=} \mathcal{N}(0, \lambda^2)$$

is a white noise including both the measurement error and the residual variability affecting the system [1]. By combining Equations (3.1) and (3.2), the statistical equation which

1. see Kennedy and O'Hagan (2001) for details about residual variability

links the code output with the field measurement is

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i \tag{3.3}$$

where a zero-mean Gaussian process prior distribution on $b(.)$ is defined as follows:

$$b(.) \sim \mathcal{GP}(0, \sigma_b^2 \Sigma_{\boldsymbol{\Psi}_b}(.,.)), \tag{3.4}$$

where σ_b^2 is a scale parameter and $\boldsymbol{\Psi}_b$ is a vector of range and possibly additional smoothness parameters. The zero-mean means no bias is assumed between the code outputs and the reality on the input space $\mathcal{X}$. In addition, Bayarri et al. (2007) consider the restrictive case where the code is time-consuming and is thus emulated by a Gaussian process as well. The estimation of both $\boldsymbol{\theta}$ and the two Gaussian processes is conducted through a modular approach which is not fully Bayesian (Liu et al., 2009). In the one hand, the field measurements are neglected for estimating $\boldsymbol{\theta}$. In the other hand, plug in estimates of some hyperparameters are considered instead of sampling their posterior distribution (Bayarri et al., 2007).

After the estimation procedure is completed, Bayarri et al. (2007) provides a guidance on how to quantify the resulting uncertainty on $r(\mathbf{x})$. A predictor of $r(\mathbf{x})$ which is referred to as a discrepancy corrected predictor $\hat{r}(\mathbf{x})$ can be computed by integrating over the joint posterior distribution of $\boldsymbol{\theta}$ and the discrepancy function $b(\mathbf{x})$. Bayarri et al. (2007) show on a pedagogic example that the discrepancy corrected-predictions yield good results for predicting the physical system $r(\mathbf{x})$. The accuracy of $\hat{r}(\mathbf{x})$ is assessed with tolerance bounds which are similar to credibility intervals on $r(\mathbf{x})$. Such a detailed description of the uncertainties surrounding code predictions is very advantageous for engineers, over a binary response "yes or no the code is correct" because a computer code could be validated in an input domain and rejected in another, depending on how large the credibility bounds are in each of them.

However, the main issue of this method resides in the confounding between $\boldsymbol{\theta}$ and $b(\mathbf{x})$, which is apparent from Equation (3.3). This implies that the posterior distribution of $\boldsymbol{\theta}$ is likely be multimodal (Bayarri et al., 2007; Loeppky et al., 2006). In fact, any value of the parameter $\boldsymbol{\tau}$ yields a code discrepancy depending on $\boldsymbol{\tau}$:

$$b_{\boldsymbol{\tau}}(\mathbf{x}) = r(\mathbf{x}) - y_{\boldsymbol{\tau}}(\mathbf{x}). \tag{3.5}$$

In Equation (3.3), the dependency of the discrepancy function to $\boldsymbol{\theta}$ can be omitted by considering $\boldsymbol{\theta}$ as being the best fitting parameter in the regression model (3.3) after having specified a Gaussian process prior (3.4) on the code discrepancy. However, confounding with $b(.)$ can still persist as we will see in Section 3.5. The easier way of avoiding this problem will be to consider the unbiaised equation (Cox et al., 2001):

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + \epsilon_i. \tag{3.6}$$

However, Bayarri et al. (2007) provides some arguments in favor of considering the discrepancy function:

- preventing over-fitting: if $\boldsymbol{\theta}$ is estimated from (3.6), the estimation of $\boldsymbol{\theta}$ could compensate (hide) a discrepancy, which must be avoided;
- the discrepancy-corrected predictor is stable although the discrepancy predictor $\hat{b}(\mathbf{x})$ and the posterior distribution of $\boldsymbol{\theta}$ are highly correlated and sensitive to the prior distributions.

In our mind, these arguments are not completely convincing. Indeed, if there is a discrepancy between the code and the physical system, a formal method should be carried out to detect it. This information could be fundamental to convince numerical practitioners to use discrepancy-corrected predictions. Conversely, if there is no discrepancy, then why bother unnecessarily complicating the statistical model for code validation? It would be very important to know that because the computer code could be directly used for predicting the physical system instead of the corrected predictor.

In the next section, we recall the Bayesian strategy of model selection to test whether or not the discrepancy function should be taken into account in the validation procedure. This idea was originally suggested for code validation in an earlier technical report dealing with this framework (Bayarri et al., 2002). Curiously, no feedback is provided in the published paper (Bayarri et al., 2007).

3.3 Bayesian model selection

The term $b(\mathbf{x})$ in the statistical equation (3.3) can be controversial for numerical practitioners who aim at constructing computer codes the closest possible to the reality. In the last section, we have argued in favor of testing whether or not there is a significant gap between the code output and the physical system before considering a discrepancy such a term. We begin by recalling the method for Bayesian model selection.

Let us consider the problem of choosing between two statistical models $\mathcal{M}_0$ depending on a vector of parameters $\mathbf{p}_0$ and $\mathcal{M}_1$ depending on a vector of parameters $\mathbf{p}_1$. The Bayes factor is calculated as the ratio of the *posterior* probabilities of each model multiplied by the ratio of *prior* probabilities of each model. Hence, it can be written as the ratio of marginal likelihoods:

$$\begin{aligned} \mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f) &:= \frac{\mathrm{P}(\mathcal{M}_0|\mathbf{z}^f)}{\mathrm{P}(\mathcal{M}_1|\mathbf{z}^f)} \times \frac{\mathrm{P}(\mathcal{M}_1)}{\mathrm{P}(\mathcal{M}_0)} \\ &= \frac{\int_{\mathbf{p}_0} f_0(\mathbf{z}^f|\mathbf{p}_0)\pi_0(\mathbf{p}_0)d\mathbf{p}_0}{\int_{\mathbf{p}_1} f_1(\mathbf{z}^f|\mathbf{p}_1)\pi_1(\mathbf{p}_1)d\mathbf{p}_1} \end{aligned}$$

where $f_i(.)$ and $\pi_i(.)$ are respectively the likelihood function and the prior distribution. From the Bayes factor, one can easily obtain the posterior probability of each model,

$$\mathrm{P}(\mathcal{M}_0|\mathbf{z}^f) = \frac{\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)}{\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f) + \frac{\mathrm{P}(\mathcal{M}_1)}{\mathrm{P}(\mathcal{M}_0)}}$$

and then $\mathrm{P}(\mathcal{M}_1|\mathbf{z}^f) = 1 - \mathrm{P}(\mathcal{M}_0|\mathbf{z}^f)$.

Since the marginal likelihoods strongly depend on the prior assumptions $\pi_1(\mathbf{p}_1)$ and $\pi_0(\mathbf{p}_0)$, they should be carefully elicited. From here, two situations can be distinguished. The first is where some prior information is available on both $\mathbf{p}_0$ and $\mathbf{p}_1$. Then, it is recommended to build compatible prior distributions to avoid favoring the model $\mathcal{M}_0$ over the model $\mathcal{M}_1$ and vice-versa (Celeux et al., 2006). The second consists in performing an objective Bayesian model selection (Casella and Moreno, 2006).

Unfortunately, unlike the posterior distribution which can be well defined even if an improper prior is used, the Bayes factor is defined only in the case of a proper prior. This is because improper priors are defined *up to an arbitrary factor*. For instance, the uniform prior on $\mathbf{p}_1$ is defined as: $\pi_1(\mathbf{p}_1) \propto 1$, which we can write: $\pi_1(\mathbf{p}_1) = c_1$, with c_1 an

arbitrary positive number. More generally, $\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)$ is then defined up to an arbitrary multiplicative ratio c_0/c_1 where c_1 is a constant specific to $\pi_1(\mathbf{p}_1) := c_1 g_1(\mathbf{p}_1)$ and c_0 is a constant specific to $\pi_0(\mathbf{p}_0) := c_0 g_0(\mathbf{p}_0)$.

However, some ingenious techniques have been developed to circumvent this limit, such as the fractional Bayes factor introduced by O'Hagan (1995) and the intrinsic Bayes factor (IBF) introduced by Berger and Pericchi (1996). The idea advanced by these authors consists in updating the improper priors using a training set of the data $\mathbf{z}^f(m) \subset \mathbf{z}^f$ containing only $m < n$ field measurements, then using the resulting proper (partial) posterior distribution instead of the improper prior distribution in the Bayes factor expression. More formally, we have for $i = 0, 1$,

$$\pi_i(\mathbf{p}_i|\mathbf{z}^f(m)) = \frac{f_i(\mathbf{z}^f(m)|\mathbf{p}_i)\pi_i(\mathbf{p}_i)}{m_i(\mathbf{z}^f(m))}. \tag{3.7}$$

where

$$m_i(\mathbf{z}^f(m)) = \int f_i(\mathbf{z}^f(m)|\mathbf{p}_i)\pi_i(\mathbf{p}_i)\mathrm{d}\mathbf{p}_i.$$

Let $\mathbf{z}^f(-m) := \mathbf{z}^f \backslash \mathbf{z}^f(m)$ be the remaining data. Then,

$$\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f(-m)|\mathbf{z}^f(m)) := \frac{\int f_0(\mathbf{z}^f(-m)|\mathbf{z}^f(m),\mathbf{p}_0)\pi_0(\mathbf{p}_0|\mathbf{z}^f(m))\mathrm{d}\mathbf{p}_0}{\int f_1(\mathbf{z}^f(-m)|\mathbf{z}^f(m),\mathbf{p}_1)\pi_1(\mathbf{p}_1|\mathbf{z}^f(m))\mathrm{d}\mathbf{p}_1} \tag{3.8}$$

$$= \frac{\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)}{\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f(m))}. \tag{3.9}$$

The resulting expression (3.8) is known as a partial Bayes Factor. Equation (3.9) is obtained by injecting the right member of Equation (3.7) under the integral in Equation (3.8). Because it depends on both the size m and the choice of the training sample $\mathbf{z}^f(m)$, Berger and Pericchi (1996) have advised to take m as the smallest value making proper $\pi_i(\mathbf{p}_i|\mathbf{z}^f(m))$, then to compute the average of the partial Bayes factor over all the subsets $z_f(m)$ of size m. It is referred to as the arithmetic intrinsic Bayes Factor:

$$\begin{aligned}\mathrm{B}^{\mathrm{A}}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f(-m)) &= \mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f) \times \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}=\mathrm{C}(n,m)} \left(\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f(m_i))\right)^{-1} \\ &= \mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f) \times \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}=\mathrm{C}(n,m)} \left(\mathrm{B}_{\mathcal{M}_1,\mathcal{M}_0}(\mathbf{z}^f(m_i))\right)\end{aligned} \tag{3.10}$$

where $\mathrm{C}(n,m)$ is the number of m-combinations from the given set of the n field measurements. In Equation (3.10), the dependency on both constants c_0 and c_1 is removed. Instead of arithmetic averaging, Berger and Pericchi (1996) also consider the geometric intrinsic Bayes Factor where a geometric averaging is performed. The strengths and weaknesses of both arithmetic and geometric IBF are discussed in Berger and Pericchi (1996). From now on, we will focus on the arithmetic IBF which is recommended by the authors. In addition, we will see in Section 3.4, considering either of averaging gives the same for a special case.

3.4 Application to code validation

Let us assume that the computer code is linear with respect to $\boldsymbol{\theta}$, we have

$$y_{\boldsymbol{\theta}}(\mathbf{x}) = h(\mathbf{x})^{\mathrm{T}}\boldsymbol{\theta}, \tag{3.11}$$

where $h(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_p(\mathbf{x}))^{\mathrm{T}}$ is the p-dimensional vector of the regression functions at an input location $\mathbf{x}$. According to (3.11), the statistical model (3.6) can be rewritten as

$$\mathscr{M}_0 : z_i^f = h(\mathbf{x}_i^f)^{\mathrm{T}}\boldsymbol{\theta}_0 + \epsilon_i^0, \tag{3.12}$$

where

$$\epsilon_i^0 \underset{i.i.d.}{\sim} \mathscr{N}(0, \lambda_0^2).$$

Likewise, the statistical model (3.3) can be rewritten as

$$\mathscr{M}_1 : z_i^f = h(\mathbf{x}_i^f)^{\mathrm{T}}\boldsymbol{\theta}_1 + b(\mathbf{x}_i^f) + \epsilon_i^1, \tag{3.13}$$

where

$$\epsilon_i^1 \underset{i.i.d.}{\sim} \mathscr{N}(0, \lambda_1^2),$$

and

$$b(.) \sim \mathscr{GP}(0, \sigma^2 \Sigma_{\boldsymbol{\Psi}}(.,.)), \tag{3.14}$$

where the variance-covariance matrix of $b(.)$ depends on a scale parameter σ^2 and a correlation function indexed by a vector of parameters $\boldsymbol{\Psi}$. Bayarri et al. (2007) use an exponential function:

$$\Sigma_{\boldsymbol{\theta},\nu}(\mathbf{x},\mathbf{x}') = \exp \sum_{j=1}^{d} -\frac{|x_j - x'_j|^{\nu_j}}{\theta_j} \tag{3.15}$$

where $\boldsymbol{\nu} \in]0,2]^d$ and $\boldsymbol{\theta} \in \mathbb{R}_+^d$. Here, $\boldsymbol{\Psi} = (\boldsymbol{\theta}, \boldsymbol{\nu})$ includes a vector of regularity parameters $\boldsymbol{\nu}$ and a vector of range parameters $\boldsymbol{\theta}$. The function (3.15) is called separable, because it can be written as a product of one-dimensional correlation functions. When $\boldsymbol{\nu} = (2, \cdots, 2)$, (3.15) is the separable Gaussian kernel. In Section 3.5, simulations will be done using an isotropic Gaussian correlation

$$\Sigma_{\Psi}(\mathbf{x},\mathbf{x}') = \exp -\frac{||\mathbf{x} - \mathbf{x}'||^2}{\Psi}, \tag{3.16}$$

where $\Psi \in \mathbb{R}^+$ is a single range parameter meaning that the correlation length is assumed unchanged whatever the d-coordinates.

Remark In Equation (3.14), $b(.)$ should be understood as a random error term, rather than a a deterministic discrepancy function, thus avoiding the identifiability problem arising from Equation (3.1). Despite all, confounding between $\boldsymbol{\theta}_1$ and $(\sigma^2, \boldsymbol{\Psi})$ can still occur in Model (3.13) (see Section 3.5).

Let $\mathbf{p}_1 := (\boldsymbol{\theta}_1, \boldsymbol{\Psi}, \sigma^2, k)$ be the unknown vector of parameters in model $\mathscr{M}_1$, where $k := \lambda_1^2\sigma^{-2}$ is the noise-to-signal ratio. The likelihood function of $\mathbf{p}_1$ in $\mathscr{M}_1$ is given by

$$f_1(\mathbf{z}^f|\mathbf{p}_1) = \frac{|\mathrm{V}_{k,\boldsymbol{\Psi}}|^{-1/2}}{(2\pi)^{n/2}(\sigma^2)^{n/2}} \exp -\frac{1}{2\sigma^2}\big(\mathbf{z}^f - h(\mathbf{X}^f)\boldsymbol{\theta}_1\big)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}\big(\mathbf{z}^f - h(\mathbf{X}^f)\boldsymbol{\theta}_1\big) \tag{3.17}$$

where $\mathrm{V}_{k,\boldsymbol{\Psi}}(i,j) := \Sigma_{\boldsymbol{\Psi}}(\mathbf{x}_i,\mathbf{x}_j) + k\mathrm{I}_n$ and $h(\mathbf{X}^f)$ is the design matrix of dimension $n \times p$ which is written as

$$h(\mathbf{X}^f) = [h(\mathbf{x}_1^f), \cdots, h(\mathbf{x}_n^f)]^{\mathrm{T}}.$$

Let $\mathbf{p}_0 := (\boldsymbol{\theta}_0, \lambda_0^2)$ be the unknown vector of parameters with respect to $\mathscr{M}_0$. The likelihood function of $\mathbf{p}_0$ in $\mathscr{M}_0$ is given by

$$f_0(\mathbf{z}^f|\mathbf{p}_0) = \frac{1}{(2\pi)^{n/2}(\lambda_0^2)^{n/2}} \exp -\frac{1}{2\lambda_0^2}\big(\mathbf{z}^f - h(\mathbf{X}^f)\boldsymbol{\theta}_0\big)^{\mathrm{T}}\big(\mathbf{z}^f - h(\mathbf{X}^f)\boldsymbol{\theta}_0\big). \tag{3.18}$$

Computing the Bayes factor for selecting between $\mathscr{M}_0$ and $\mathscr{M}_1$ requires to specify a *prior* distribution on both $\mathbf{p}_0$ and $\mathbf{p}_1$. General recommendations on this point are summarized in Berger et al. (2001). Either genuine prior information is available and should be used in constructing the prior distribution or no prior information is available and a so-called default prior should be specified (Kass and Wasserman, 1996). A vague prior could be used as an alternative such as the conjugate Inverse-gamma-Gaussian prior on $(\boldsymbol{\theta}_0, \lambda_0^2)$ and $(\boldsymbol{\theta}_1, \sigma^2)$. Unfortunately, the hyper-parameters of these priors are often chosen arbitrarily, even though the marginal likelihood is known to be sensitive to their specification. In addition, using informative priors in computing the Bayes factor requires the priors in each model to be compatible in some sense (Consonni et al., 2008).

In this paper, we consider the general form of prior below:

$$\begin{aligned}\pi(\mathbf{p}_1) &= \pi(\boldsymbol{\theta}_1, \sigma^2, \boldsymbol{\Psi}|k)\pi(k) \\ &\propto \frac{\pi(\boldsymbol{\Psi}|k)}{(\sigma^2)^a}\pi(k)\end{aligned} \tag{3.19}$$

and

$$\pi(\mathbf{p}_0) \propto \frac{1}{(\lambda_0^2)^a} \tag{3.20}$$

In the case where k is known, the prior (3.19) has been deeply studied. Berger et al. (2001) and Paulo (2005) have established that most used default priors such as the Jeffreys prior, the independence Jeffreys prior [2], and the reference prior are special cases of (3.19). The last part of their work was about whether or not the resulting posterior is proper. Paulo (2005) has proved that all of theses priors have a proper posterior distribution in the multidimensional case under some assumptions including the *separability* of the correlation function. In this paper, we assume that the correlation function is isotropic, meaning that it only depends on $||\mathbf{x}-\mathbf{x}'||$. In this case, Berger et al. (2001) has proved that the independence Jeffreys prior takes the form (3.19) with

$$a = 1 \text{ and } \pi(\Psi|k) \propto \left\{\mathrm{tr}[\mathrm{U}_\Psi^2] - \frac{1}{n}\mathrm{tr}[\mathrm{U}_\Psi]^2\right\}^{\frac{1}{2}} \tag{3.21}$$

where $\mathrm{U}_\Psi = \frac{\partial \mathrm{V}_{k,\Psi}}{\partial \Psi}\mathrm{V}_{k,\Psi}^{-1}$. The Jeffreys-rule prior is also of the form (3.19) with

$$a = 1 + \frac{p}{2} \text{ and } \pi(\Psi|k) \propto |h(\mathbf{X}^f)\Sigma_\Psi^{-1}h(\mathbf{X}^f)|^{\frac{1}{2}}\left\{\mathrm{tr}[\mathrm{U}_\Psi^2] - \frac{1}{n}\mathrm{tr}[\mathrm{U}_\Psi]^2\right\}^{\frac{1}{2}}. \tag{3.22}$$

Finally, we consider the Berger-Bernardo reference prior of the form

$$a = 1 \text{ and } \pi(\Psi|k) \propto \left\{\mathrm{tr}[\mathrm{W}_\Psi^2] - \frac{1}{n}\mathrm{tr}[\mathrm{W}_\Psi]^2\right\}^{\frac{1}{2}}, \tag{3.23}$$

where $\mathrm{W}_\Psi = \frac{\partial \mathrm{V}_{k,\Psi}}{\partial \Psi}\mathrm{V}_{k,\Psi}^{-1}\mathrm{P}_\Psi$ and $\mathrm{P}_\Psi = \mathrm{I} - h(\mathbf{X}^f)(h(\mathbf{X}^f)\mathrm{V}_{k,\Psi}^{-1}h(\mathbf{X}^f))^{-1}h(\mathbf{X}^f)\mathrm{V}_{k,\Psi}^{-1}$. Berger et al. (2001) proved that (3.22) is proper whereas (3.21) and (3.23) can be proper or improper depending on the form of $h(\mathbf{X}^f)$ (see Section 3.5).

The marginal likelihood of Model $\mathscr{M}_1$ can be rewritten as

$$\int_{\mathbf{p}_1} f_1(\mathbf{z}^f|\mathbf{p}_1)\pi_1(\mathbf{p}_1)\mathrm{d}\mathbf{p}_1 = \int_k \int_{\boldsymbol{\Psi}} \tilde{f}_1(\mathbf{z}^f|\mathbf{p}_1)\pi_1(k)\pi_1(\boldsymbol{\Psi}|k)\mathrm{d}k\mathrm{d}\boldsymbol{\Psi}, \tag{3.24}$$

2. which is obtained by assuming $\boldsymbol{\theta}$ and $(\boldsymbol{\Psi}, \sigma^2, k)$ are independent a priori

where

$$\tilde{f}_1(\mathbf{z}^f|\mathbf{p}_1) \propto \frac{(2\pi)^{\frac{p-n}{2}}}{(|h(\mathbf{X}^f)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}h(\mathbf{X}^f)||\mathrm{V}_{k,\boldsymbol{\Psi}}|)^{1/2}} \times \frac{\Gamma(\frac{n-p}{2}-1+a)}{\mathrm{Q}_{k,\boldsymbol{\Psi}}^{(\frac{n-p}{2}-1+a)}},$$

with $\mathrm{Q}_{k,\boldsymbol{\Psi}}$ the generalized residuals sum of squares

$$\begin{aligned}\mathrm{Q}_{k,\boldsymbol{\Psi}} &= \frac{1}{2}\mathbf{f}_{\mathbf{M}}{}^{\mathrm{T}}\left(\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1} - \mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}h(\mathbf{X}^f)\left(h(\mathbf{X}^f)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}h(\mathbf{X}^f)\right)^{-1}h(\mathbf{X}^f)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}\right)\mathbf{z}^f \\ &= \left(\mathbf{z}^f - h(\mathbf{X}^f)\hat{\boldsymbol{\theta}}_1\right)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}\left(\mathbf{z}^f - h(\mathbf{X}^f)\hat{\boldsymbol{\theta}}_1\right),\end{aligned}$$

where

$$\hat{\boldsymbol{\theta}}_1 := \left(h(\mathbf{X}^f)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}h(\mathbf{X}^f)\right)^{-1}h(\mathbf{X}^f)^{\mathrm{T}}\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}.$$

Since $\mathscr{M}_0$ and $\mathscr{M}_1$ are nested models, the marginal likelihood of $\mathscr{M}_0$ is explicit because the parameters (σ^2, k) in Equation (3.17) play the role of (σ^2, λ_1^2) in Equation (3.18) by setting $\mathrm{V}_{k,\boldsymbol{\Psi}} = \mathrm{I}_n$. Hence,

$$\int_{\mathbf{p}_0} f_0(\mathbf{z}^f|\mathbf{p}_0)\pi_0(\mathbf{p}_0)d\mathbf{p}_0 \propto \frac{(2\pi)^{\frac{p-n}{2}}}{(|h(\mathbf{X}^f)^{\mathrm{T}}h(\mathbf{X}^f)|)^{1/2}} \times \frac{\Gamma(\frac{n-p}{2}-1+a)}{\mathrm{Q}^{(\frac{n-p}{2}-1+a)}} \tag{3.25}$$

where

$$\begin{aligned}\mathrm{Q} &= \frac{1}{2}\mathbf{z}^{f^{\mathrm{T}}}\left(\mathrm{I} - h(\mathbf{X}^f)\left(h(\mathbf{X}^f)^{\mathrm{T}}h(\mathbf{X}^f)\right)^{-1}h(\mathbf{X}^f)^{\mathrm{T}}\right)\mathbf{z}^f \\ &= \left(\mathbf{z}^f - h(\mathbf{X}^f)\hat{\boldsymbol{\theta}}_0\right)^{\mathrm{T}}\left(\mathbf{z}^f - h(\mathbf{X}^f)\hat{\boldsymbol{\theta}}_0\right)\end{aligned}$$

with

$$\hat{\boldsymbol{\theta}}_0 = \left(h(\mathbf{X}^f)^{\mathrm{T}}h(\mathbf{X}^f)\right)^{-1}h(\mathbf{X}^f)^{\mathrm{T}}\mathbf{z}^f. \tag{3.26}$$

Contrary to Section 3.3 where $\mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(.)$ has been presented, we now introduce the expression of $\mathrm{B}_{\mathscr{M}_1,\mathscr{M}_0}(.)$. It is given by

$$\mathrm{B}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f) = \int_k \int_{\boldsymbol{\Psi}} \mathrm{LB}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f|k,\boldsymbol{\Psi})\pi(\boldsymbol{\Psi}|k)\pi(k)\mathrm{d}k\mathrm{d}\boldsymbol{\Psi} \tag{3.27}$$

where

$$\mathrm{LB}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f|k,\boldsymbol{\Psi}) = \frac{|h(\mathbf{X}^f)^{\mathrm{T}}h(\mathbf{X}^f)|^{1/2}\mathrm{Q}^{(\frac{n-p}{2}-1+a)}}{(|h(\mathbf{X}^f)\mathrm{V}_{k,\boldsymbol{\Psi}}^{-1}h(\mathbf{X}^f)||\mathrm{V}_{k,\boldsymbol{\Psi}}|)^{1/2}\mathrm{Q}_{k,\boldsymbol{\Psi}}^{(\frac{n-p}{2}-1+a)}} \tag{3.28}$$

is the local Bayes factor (Sinharay and Stern, 2002). It is ill-defined from the discussion in Section 3.3.

Now, we point out an important equality between the IBF and the standard Bayes factor under some assumptions concerning the prior distribution (3.19).

Proposition 3.4.0.1 *Let us assume $a = 1$, $\pi(\boldsymbol{\Psi}, k)$ is proper and $m = p + 1$. Then,*

$$\mathrm{B}^{\mathrm{A}}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f(-m)) = \mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f) \tag{3.29}$$

where the dependency of $\mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f)$ on both constants c_0 and c_1 is removed because $\mathrm{B}^{\mathrm{A}}_{\mathscr{M}_0,\mathscr{M}_1}(.)$ does not depend on them anymore (see equation 3.10).

Proof The proof consists in writing

$$\mathrm{B}^{\mathrm{A}}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f(-m)) = \mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f) \times \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \mathrm{B}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f(m(i))) \tag{3.30}$$

$$= \mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f) \times \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \int_k \int_{\boldsymbol{\Psi}} \mathrm{LB}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f(m_i)|k,\boldsymbol{\Psi})\pi(\boldsymbol{\Psi}|k)\pi(k)\mathrm{d}k\mathrm{d}\boldsymbol{\Psi}. \tag{3.31}$$

When $a = 1$ in Equation (3.19), that is $\pi(\boldsymbol{\theta},\sigma^2) = \sigma^{-2}$ is the right-Haar prior, we have $\mathrm{LB}_{\mathscr{M}_1,\mathscr{M}_0}(\mathbf{z}^f(m)|k,\boldsymbol{\Psi}) = 1$ for any minimal training sample $\mathbf{z}^f(m)$ because the model $\mathscr{M}_0$ and $\mathscr{M}_1$ exhibit the same invariant structure, up to the individuals parameters $\boldsymbol{\Psi}$ and k attached to $\mathscr{M}_1$ (Berger et al., 1998). Hence,

$$\mathrm{B}^{\mathrm{A}}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f(-m)) = \mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f) \times \frac{1}{\mathrm{N}}\sum_{i=1}^{\mathrm{N}} \int_k \int_{\boldsymbol{\Psi}} \pi(\boldsymbol{\Psi}|k)\pi(k)\,dk\,d\boldsymbol{\Psi}$$

$$= \mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f) \tag{3.32}$$

because the prior distribution $\pi(\boldsymbol{\Psi},k)$ is assumed to be proper.

The standard Bayes factor $\mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f)$ can be computed much faster than $\mathrm{B}^{\mathrm{A}}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f(-m))$ because it requires one integration over k and $\boldsymbol{\Psi}$ compared to $\mathrm{N}+1$, which represents a substantial reduction in computation time when $\mathrm{N} = \mathrm{C}(n,m)$ is large.

In the simulation study (see section 3.5), we will use the correlation function (3.16) and we will compare the results of our model selection procedure for each of the prior distributions below:

1. a uniform/beta prior defined by $\Psi \sim \mathscr{U}[x_{min}, x_{max}]$ and $k \sim \mathscr{B}(a_k, b_k)$, which is clearly proper;
2. the reference prior (3.23) which is proven to be proper when $\mathbf{1}$ is a column of $h(\mathbf{X}^f)$ (Berger et al., 2001) and $k \sim \mathscr{B}(a_k, b_k)$;
3. the independence Jeffreys prior (3.21) which is proven to be proper when $\mathbf{1}$ is not a column of $h(\mathbf{X}^f)$ (Berger et al., 2001) and $k \sim \mathscr{B}(a_k, b_k)$.

Therefore, the first prior distribution is proper; the second prior distribution is proper when $\mathbf{1}$ is a column of $h(\mathbf{X}^f)$; and the third prior distribution is proper when $\mathbf{1}$ is not a column of $h(\mathbf{X}^f)$. In fact, we were able to verify using numerical integration that all of them are proper whether or not $\mathbf{1}$ is a column of $h(\mathbf{X}^f)$ (although no proof of that has been established in this paper). According to the proposition 3.4.0.1, the IBF could be thus computed using $\mathrm{B}_{\mathscr{M}_0,\mathscr{M}_1}(\mathbf{z}^f)$ where the marginal likelihood of the model $\mathscr{M}_1$ (3.24) boils down to a two-dimensional integral due to the choice of an isotropic correlation.

Remark In the simulation study, we will not use the Jeffreys-rule prior (3.22) because it does not satisfy the conditions of the proposition 3.4.0.1 ($a = 1 + \frac{p}{2}$ instead of $a = 1$).

3.5 Simulation study

We have chosen $a_k = 1$ and $b_k = 3$, meaning that the scale of the noise is expected to be smaller than the scale of the code discrepancy. We compare the results of the testing procedure on simulated data $\mathbf{z}^f$. They are simulated according the model $\mathscr{M}_1$ where $\mathbf{X}^f = \{\frac{i}{n}\}_{i=1}^n$ is equally-spaced on $[0,1]$. Several configurations are tested:

— a constant code and $\theta_1 = 1$,
— a linear code in x, that is $h(x) = (1, x)$ and $\boldsymbol{\theta}_1 = (1, 1)$,
— a degree 2 polynomial code in x, that is $h(x) = (1, x, x^2)$ and $\boldsymbol{\theta}_1 = (1, 1, 1)$.

In all three cases, we have set $\sigma^2 = 0.1$ and $\lambda^2 = 0.01$. Figure 3.2, 3.3 and 3.4 display the value of the Bayes factor $B_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)$ for $n = 30$ against $\boldsymbol{\Psi}$ varying in $[0, 1]$ with respect to the three prior distributions above. It is computed with a numerical integration routine from the library R cubature. Boxplots are constructed with 100 Bayes factor values.

Confounding between the code and the correlation We can expect the larger the correlation length Ψ, the smaller the Bayes Factor and thus favoring the model $\mathcal{M}_1$. Instead, Figure 3.2, 3.3 and 3.4 show that the larger the correlation length, the larger the Bayes factor. If we take a close look at Figures 3.2, 3.3 and 3.4, we can see when $\Psi = 0$ (that is $\mathbf{z}^f$ are simulated from the model $\mathcal{M}_0$), the Bayes factor exceeds 1 favoring the model $\mathcal{M}_0$ as expected. As Ψ increases, we can see the Bayes factor falls near 0 favoring the model $\mathcal{M}_1$ as still expected, then rose above 1 favoring the model $\mathcal{M}_0$, which is surprising at first sight. This behavior is even more important when the number of regressions function of the code is increased because then the correlation simulated from the simulation stage can be more easily absorbed in the estimation of $\boldsymbol{\theta}_1$.

In fact, there is a severe confounding between the discrepancy function and the computer code, which is related with the delicate issue of the consistency of the Gaussian process parameters in a fixed asymptotic domain (Bachoc, 2013; Stein, 1999). In Figure 3.1, we can see the correlation is detected when $\Psi = 0.05$ because the estimated mean looks close to the real mean whereas the large correlation remains undetected when $\Psi = 0.8$ because the estimated mean goes through the measurements. Therefore, $\boldsymbol{\theta}$ has to be interpreted as the best-fitting parameter which can be strongly different from the true value, that is the one used for the simulation stage.

Role of the prior distribution We can see the uniform/beta prior yields the lowest values of the Bayes factor when the field measurements $\mathbf{z}^f$ are simulated from the model $\mathcal{M}_0$ ($\Psi = 0$). No large differences can be noted between the reference/beta prior and the independence Jeffreys/beta prior, except the reference/beta prior may yield larger values of the Bayes factor than the independence Jeffreys/beta prior (we can see some values of the Bayes factor over 10 when $\Psi = 0$ for the reference/beta prior).

Figure 3.1 – *Dots are* $\mathbf{z}^f$*, simulated from the model* $\mathcal{M}_1$ *with a degree* 2 *polynomial code in* x *(solid line) plus a zero mean-GP where* $\Psi = 0.05$ *(left) and* $\Psi = 0.8$ *(right). The dotted line is the posterior mean of the code which is estimated from the model* $\mathcal{M}_1$ *with the reference/beta prior.*

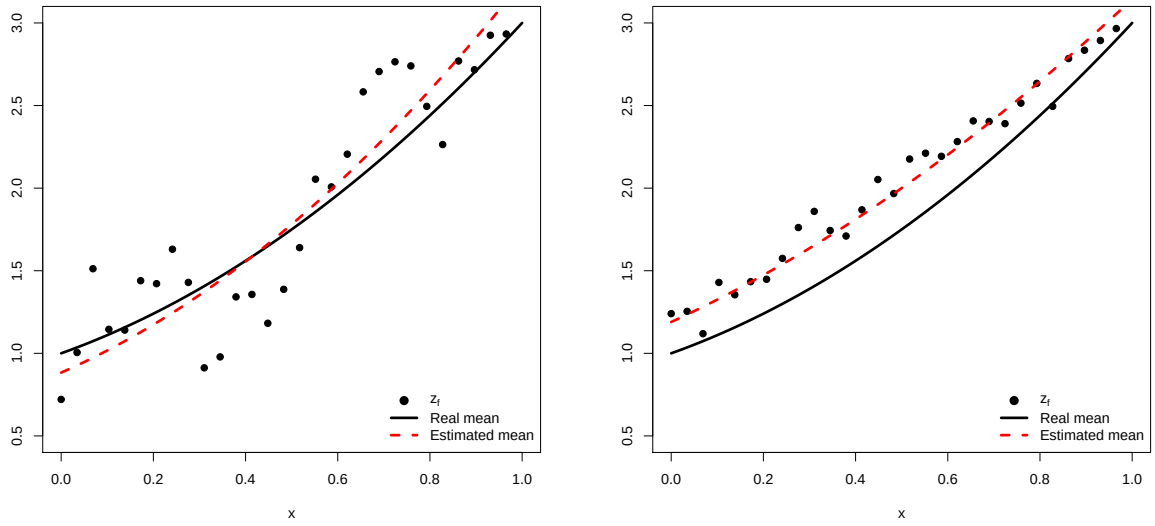


Figure 3.2 – *Bayes factor* $\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)$ *plotted against the value of* Ψ *with the uniform/beta prior. Left:* $\mathbf{z}^f$ *are simulated with a constant code. Middle:* $\mathbf{z}^f$ *are simulated with a linear code in* x*. Right:* $\mathbf{z}^f$ *are simulated with a degree* 2 *polynomial code in* x*.*

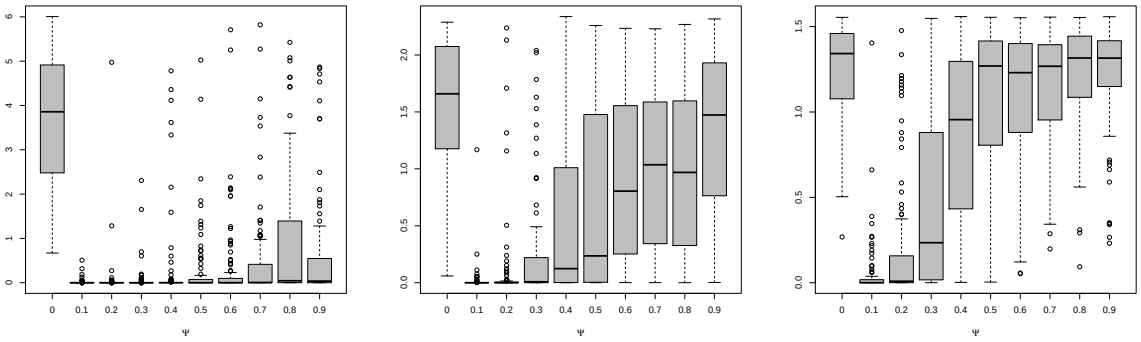


Figure 3.3 – *Bayes factor* $\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)$ *plotted against the value of* Ψ *with the reference/beta prior. Left:* $\mathbf{z}^f$ *are simulated with a constant code. Middle:* $\mathbf{z}^f$ *are simulated with a linear code in* x*. Right:* $\mathbf{z}^f$ *are simulated with a degree* 2 *polynomial code in* x*.*

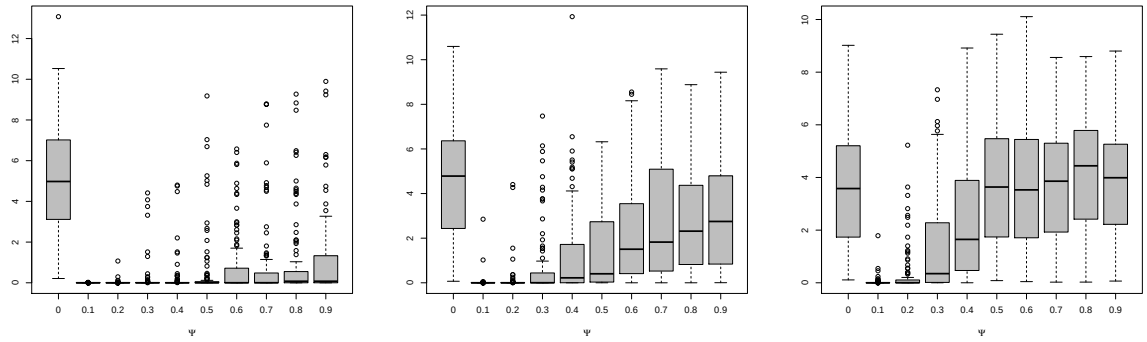
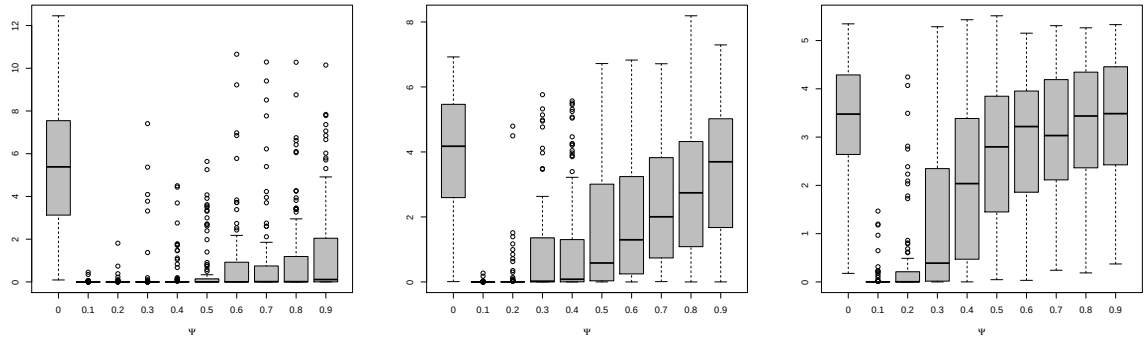


Figure 3.4 – *Bayes factor* $\mathrm{B}_{\mathcal{M}_0,\mathcal{M}_1}(\mathbf{z}^f)$ *plotted against the value of* Ψ *with the independence Jeffreys/beta prior. Left:* $\mathbf{z}^f$ *are simulated with a constant code. Middle:* $\mathbf{z}^f$ *are simulated with a linear code in* x*. Right:* $\mathbf{z}^f$ *are simulated with a degree* 2 *polynomial code in* x*.*



3.6 Validation of an industrial code for power plant production control

Our Bayes factor methodology is now applied to the validation of an actual industrial computer code which is used to predict the productivity of an electric power plant based on a series of measurements (of temperature, pressure, and flow among others) acquired by specific sensors installed throughout the plant. Such predictions are compared on a weekly basis to the actual power produced by the plant. When the latter is significantly lower than expected, the difference is attributed to either ageing or failure of one or several components. The computer code is then used to diagnose the probable causes of the observed loss of power. Hence, validation of the computer code is crucial to ensure an early detection of power decrease and a correct identification of its causes.

The code was written in Modelica [3] which is an object-oriented, equation-based language. It is well adapted to describe complex systems, characterized by the interaction of many components. The version considered for this study has $d = 20$ input variables ($\mathbf{x} \in \mathbb{R}^d$), corresponding to the aforementioned measurements of temperature, pressure, etc. and two outputs $(y_\theta^s(\mathbf{x}), y_\theta^w(\mathbf{x}))$. These are predictions of the condenser pressure $r^s(\mathbf{x})$ and the electric power $r^w(\mathbf{x})$ produced by the plant, for an overall computation time of approximately 30 seconds.

After conducting a sensitivity analysis, the code has been shown to have $p = 2$ influential parameters $\boldsymbol{\theta} = (\theta_1, \theta_2) \in \mathbb{R}^2$ which consist of the heat transfer coefficient of the condenser θ_1 and the isentropic yield of the main turbine θ_2.

Prior knowledge is available for $\boldsymbol{\theta}$, consisting of a nominal value $\boldsymbol{\theta}^\star = (\theta_1^\star, \theta_2^\star)$, along with plausible upper and lower bounds $\boldsymbol{\theta}^\star \pm \boldsymbol{\Delta}$.

The available field measurements $\left(\mathbf{X}^f, \mathbf{z}^f = (\mathbf{z}^s, \mathbf{z}^w)\right)$ of size $n = 24$ come from weekly tests conducted at the beginning of the plant's life cycle. They are in fact representative of the *reference* behavior of the plant, that is, prior to failure and ageing of its components.

Linear approximation The code can be well approximated by a linear response surface which is obtained by a first-order Taylor approximation near the nominal value. More specifically, by conducting a preliminary sensitivity analysis using the Morris method (Morris (1991)), we found that each of the two code outputs evaluated at any observed input location $\mathbf{x}_i^f$ can be calculated as

$$y_{\boldsymbol{\theta}}^s(\mathbf{x}_i^f) \approx y_{\boldsymbol{\theta}^\star}^s(\mathbf{x}_i^f) + h^s(\mathbf{x}_i^f)^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^\star), \tag{3.33}$$

and

$$y_{\boldsymbol{\theta}}^w(\mathbf{x}_i^f) \approx y_{\boldsymbol{\theta}^\star}^w(\mathbf{x}_i^f) + h^w(\mathbf{x}_i^f)^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^\star), \tag{3.34}$$

where

$$h^s(\mathbf{x}_i^f) = \nabla_{\boldsymbol{\theta}} y_{\boldsymbol{\theta}^\star}(\mathbf{x}_i^f)$$

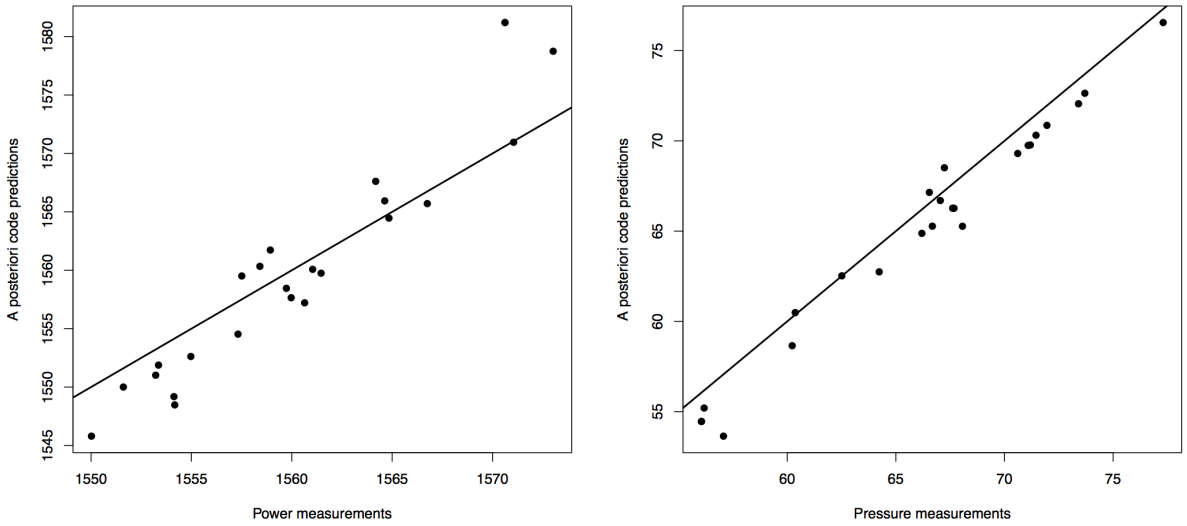
is the vector of partial derivatives of the pressure with respect to each parameter and

$$h^w(\mathbf{x}_i^f) = \nabla_{\boldsymbol{\theta}} y_{\boldsymbol{\theta}^\star}(\mathbf{x}_i^f)$$

is the vector of partial derivatives of the power with respect to each parameter. They are evaluated in practice by finite differences. Hence, the linear assumption (3.11) is seen to

3. https://www.modelica.org

Figure 3.5 – *Unbiaised calibration*



hold approximately for the translated code $y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) - y_{\boldsymbol{\theta}^\star}(\mathbf{x}_i^f)$ near $\boldsymbol{\theta}^\star$. Because the preliminary sensitivity analysis has also shown that θ_1 has a negligible influence on the electric power and θ_2 has a negligible on the condenser pressure, the calibration of these two code parameters can be conducted independently. Hence, by considering θ_2 as constant in Equation (3.33) yields

$$y^s_{\theta_1}(\mathbf{x}_i^f) \approx y^s_{\theta_1^\star}(\mathbf{x}_i^f) + \frac{\partial y^s_{\theta_1}(\mathbf{x}_i^f)}{\partial \theta_1}(\theta_1 - \theta_1^\star), \tag{3.35}$$

and, likewise, considering θ_1 as constant in Equation (3.34) yields

$$y^w_{\theta_2}(\mathbf{x}_i^f) \approx y^w_{\theta_2^\star}(\mathbf{x}_i^f) + \frac{\partial y^w_{\theta_2}(\mathbf{x}_i^f)}{\partial \theta_2}(\theta_2 - \theta_2^\star). \tag{3.36}$$

Bayes factor results The Bayes factor was calculated separately for each of the two linear approximations of the code (3.35) and (3.36), to detect a possible discrepancy between $y^s_{\theta_1}(\mathbf{X}^f)$ and $r^s(\mathbf{X}^f)$, then between $y^w_{\theta_2}(\mathbf{X}^f)$ and $r^w(\mathbf{X}^f)$. To do so, we specified a right-Haar on $\pi(\theta_i, \sigma^w) \propto \frac{1}{\sigma^w}$ $(i = 1,2)$, a reference prior on Ψ conditional to k and the beta prior $\mathscr{B}(1,3)$ on k. For the pressure, the resulting Bayes factor was equal to $2.23e-18$, which is strong evidence for the existence of a code discrepancy. For the electric power, the Bayes factor was equal to 0.0034, meaning that the probability of the existance of a code discrepancy was equal to of 99.6%. Both of these values are consistent with code calibration performed from the model $\mathscr{M}_0$ where the residuals between the fitted code and the field measurements do not look like a white noise (see figure 3.5).

3.7 Conclusion

This work is motivated by the framework of Bayarri et al. (2007) dedicated to code validation where a discrepancy function is inserted between the code output and the physical system. Because this assumption is often not properly justified, it is misunderstood by engineers and numerical practitioners who are used to perform validation thanks to graphical comparisons.

In this paper, we advocate a more systematic approach, which consists in applying a statistical method to assess the gap between the code output and the physical system. A key advantage of this approach is its ability to assess the uncertainty associated with predictions of the physical system. To choose the statistical model best suited for calculating such predictions, we have proposed a Bayesian criterion to check whether or not a discrepancy should be taken in account between the code and the physical system. Several cases arise from here:

— If the discrepancy is found significant (*i.e.*, the Bayes factor $B_{\mathcal{M}_0,\mathcal{M}_1}(.)$ is close to zero), the fact this arises from a theoretically well-grounded computation, rather than an informal graphical test, should constitute a good argument to convince engineers to use discrepancy-corrected predictions,

— if the discrepancy is found not significant (*i.e.*, the Bayes factor $B_{\mathcal{M}_0,\mathcal{M}_1}(.)$ is close to one), then the code output alone can be used to predict the physical system.

— If the Bayes factor does not favor one model over another (*i.e.* $B_{\mathcal{M}_0,\mathcal{M}_1}(.)$ is close to 0.5), then Bayesian model averaging Hoeting et al. (1999) should be performed to calculate optimal predictions of the physical system.

From a more technical perspective, we have relied here on a linear assumption which makes it possible to integrate analytically over both the code parameter and the variance of the code discrepancy. In practice, this can be justified when a so-called 'best-guess' value of the code parameter can be provided by expert opinion, around which a linear approximation of the code can be constructed. For non-linear codes where no best guess can be displayed, explicit marginalization over the code parameter $\boldsymbol{\theta}$ is not feasible anymore. In this case, the Bayes factor needs to be computed in a purely numerical way, which can be a challenging issue.

In addition, we have considered an isotropic correlation, implying that the Bayes factor boils down to a two dimensional integral which can be easily computed by standard quadrature techniques. This assumption is particularly appropriate when all the input variables are of equal importance on the sensibility of the code output. In the case where the correlation is not isotropic, the Bayes factor needs to be evaluated by integrating over a higher dimensional space, with a dimension that can be as high as that of the input variable. In such cases, quadrature cannot be performed anymore, and a more efficient numerical method should be carried out.

Our last perspective is about the choice of the prior distribution which is specified on the noise-to-signal ratio. In this work, we use a beta prior which favors a noise variance smaller than the variance of the discrepancy. A relevant idea for future work could consist in calculating some default priors for $\pi(\boldsymbol{\theta},\sigma^2,\boldsymbol{\Psi},k)=\pi(\boldsymbol{\theta},\sigma^2)\pi(\boldsymbol{\Psi}|k)\pi(k)$ such as the reference prior and the Jeffreys independence prior in order to extend the seminal works of Berger et al. (2001) and Paulo (2005) when the ratio k is used.

Bibliography

AIAA (1998). Guide for the verification and validation of computational fluid dynamics simulations. *Reston VA: American Institute of Aeronautics and Astronautics*, AIAA-G-077-1998. 57

Bachoc, F. (2013). *Estimation paramétrique de la fonction de covariance dans le modèle de Krigeage par processus Gaussiens. Application à la quantification d'incertitudes en simulation numérique.* PhD thesis, Université Paris Diderot. 66

Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M., Kottas, A., Paulo, R., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2002). A framework for validation of computer models. *Tech. Rep. 128, National Institute of Statistical Sciences.* 60

Bayarri, M. J., Berger, J. O., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2):138–154. 56, 57, 58, 59, 60, 62, 71

Berger, J., De Oliveira, V., and Sanso, B. (2001). Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association*, 96(456):1361–1374. 63, 65, 71

Berger, J. and Pericchi, L. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122. 56, 61

Berger, J., Pericchi, L., and Varshavsky, J. (1998). Bayes factors and marginal distributions in invariant situations. *Sankhya: The Indian Journal of Statistics, Series A (1961-2002)*, 61(3):307–321. 65

Casella, G. and Moreno, E. (2006). Objective bayesian variable selection. *Journal of the American Statistical Association*, 101(473):157–167. 60

Celeux, G., Marin, J.-M., and Robert, C. (2006). Selection bayésienne de variables en régression linéaire. *Journal de la Société Française de Statistique*, 147(1):59–79. 60

Consonni, G., Gutierez-Pena, E., and Veronese, P. (2008). Compatible priors for bayesian model comparison with an application to the hardy-weinberg equilibrium model. *TEST*, 17(3):585–605. 63

Cox, D., Park, J., and Clifford, E. (2001). A statistical method for tuning a computer code to a data base. *Computational Statistics and Data Analysis*, 37(1):77–92. 59

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the Statistical Association*, 86(416):953–963. 57

Ferson, S., Kreinovich, V., Ginzburg, D., Myers, D., and Sentz, K. (2003). Constructing probability boxes and dempster-shafer structures. Technical report, SAND2004-3072, Albuquerque, Sandia National Laboratories. 57

Gang, H., Santner, T., and Rawlinson, J. (2009). Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics*, 51(4):464–474. 57

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417. 71

Kass, R. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91(435):1343–1370. 63

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B, Methodological*, 63(3):425–464. 57, 58

Liu, F., Bayarri, M., and Berger, J. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150. 59

Loeppky, D., Bingham, D., and Welch, W. (2006). Computer model calibration or tuning in practice. Technical report, Department of Statistics, University of British Columbia, Canada. 57, 58, 59

Morris, N. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174. 69

Oberkampf, W. and Barone, M. (2006). Measures of agreement between computation and experiment: Validation metrics. *Journal of Computational Physics*, 217(1):5–36. 57

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society. Series B*, 57(1):99–138. 61

Paulo, R. (2005). Default priors for gaussian processes. *The Annals of Statistics*, 33(2):556–582. 63, 71

Roy, C. and Oberkampf, W. (2011). A comprehensive framework for verification, validation and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2131–2144. 57

Sacks, J., W.J., W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423. 57

Santner, T., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag. 57

Sinharay, S. and Stern, H. (2002). On the sensitivty of Bayes factors to the prior distributions. *The American Statistician*, 56(3):196–201. 64

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New-York. 66

Chapter 4

Adaptive numerical designs for the calibration of computer codes

Contents

Abstract Making good predictions of a physical system using a computer code requires the inputs to be carefully specified. Some of these inputs, called control variables, have to reproduce physical conditions whereas other inputs, called parameters, are specific to the computer code and most often uncertain. The goal of statistical calibration consists in reducing their uncertainty with the help of a statistical model which links the code outputs with the field measurements. In a Bayesian setting, the posterior distribution of these parameters is typically sampled using MCMC methods. However, they are impractical when the code runs are highly time-consuming. A way to circumvent this issue consists of replacing the computer code with a Gaussian process emulator, then sampling an approximate posterior distribution based on it. Doing so, calibration is subject to an error which strongly depends on the numerical design of experiments used to fit the emulator. We aim at reducing this error by building a proper sequential design by means of the Expected Improvement criterion. Numerical illustrations in several dimensions assess the efficiency of such sequential strategies.

Keywords: Bayesian calibration, Gaussian process emulation, Expected Improvement criterion.

Résumé Le vecteur des entrées d'un code de calcul reproduisant un système physique peut être constitué à la fois de variables qui reproduisent les conditions physiques de l'expérience réelle, ainsi que de paramètres spécifiques au code de calcul (n'ayant pas d'équivalent observable dans la réalité). Lorsque les paramètres sont incertains, l'étape dite de calage consiste à mettre à jour leur incertitude à partir d'un modèle statistique reliant la réponse du code aux mesures physiques. Lorsque l'on dispose d'informations *a priori* sur la valeur de ces paramètres, il est pertinent de considérer une estimation bayésienne utilisant des méthodes MCMC. Malheureusement, lorsque les simulations sont trop coûteuses en temps d'exécution, ces méthodes ne sont plus adaptées. Un moyen pour remédier au problème consiste à substituer à la réponse du code de calcul un émulateur de type processus gaussien à partir duquel une distribution *a posteriori* approchée peut être échantillonnée par MCMC. L'erreur induite par l'échantillonnage de cette distribution à la place de la distribution *a posteriori* basée sur le code dépend fortement du plan d'expériences utilisé pour construire l'émulateur. Dans ce travail, nous réduisons cette erreur grâce à des plans d'expériences construits séquentiellement à l'aide du critère de l'amélioration espérée. Des exemples numériques illustrent l'efficacité de ces plans.

Mot-clés: Calage bayésien, Processus gaussien, Critère de l'amélioration espérée.

4.1 Introduction

This work is incorporated within the field of uncertainty quantification in computer experiments. A crucial issue in engineering (aerospace, car, nuclear, etc.) concerns the ability of computer codes (also called simulators or computer models) to mimic a physical phenomenon of interest as well as possible. In this regard, the field of so-called Verification and Validation (V&V) aims at assessing the accuracy of computer predictions for many applications. For instance, the study of V&V has become a huge preoccupation in the nuclear industry where numerical simulation is more and more used to assess the safety of installations for which physical experiments are impractical or economically unfeasible. An essential prerequisite of V&V consists in quantifying all sources of uncertainty involved in a code output (Roy and Oberkampf, 2011). Our paper is focused on the reduction of one of them, called parameter uncertainty, caused by the lack of knowledge about the value of parameters which are specific to the computer code (Kennedy and O'Hagan, 2001). They can be either non-measurable physical quantities or just tuning factors. Calibration comes down to a statistical inference of these parameters after assuming a statistical model which makes explicit the relationship between the code outputs and the field measurements (Campbell, 2006; Cox et al., 2001). Another popular framework which deals with parameter uncertainty is called History Matching (HM) (Craig et al., 1997). HM aims at detecting regions in the parameter space which appear to be incompatible with the field measurements. Based on an implausibility measure, this method is well-suited for large systems wherein the size of inputs makes immediate calibration intractable. In the same way, Sensitivity Analysis (SA) aims at detecting which parameters have a negligible effect on the code output (Saltelli et al., 2000). Hence, both HM and SA can shrink the input space before carrying out calibration.

Our work focuses on calibration in a Bayesian fashion (Bayarri et al., 2007; Kennedy and O'Hagan, 2001), rather than a frequentist one (Gramacy et al., 2014; Loeppky et al., 2006; Wong et al., 2014). This strategy of inference provides an appropriate framework to quantify the parameter uncertainty from prior to posterior distribution as new data become available. In the literature, Bayesian calibration is performed within a framework where the code predictions suffer from a systematic discrepancy for any value of parameters, which reflects the view that the mathematical equations underlying the code should not be considered as a perfect model of the real world (Higdon et al., 2004; Kennedy and O'Hagan, 2001). Even if this framework is more realistic, some confounding can appear when the parameters and the shape of the discrepancy are jointly estimated (Brynjarsdottir and O'Hagan, 2014; Loeppky et al., 2006). Because this issue is outside the scope of this paper, our presentation is centered on a statistical model which does not include the code discrepancy. However, it will be possible to generalize our framework provided the shape of the discrepancy is provided by prior expertise.

We aim at addressing Bayesian calibration when the code runs are time-consuming, a critical issue frequently arising in the field of computer experiments. Indeed, when simulations needs several hours, even several days to run, then the MCMC algorithms become unfeasible. For instance, if each simulation lasts one hour, then 10000 simulations launched by an MCMC exploration of the parameter space will require more than a year, making the calibration process impractical. A well-known solution to this issue is to replace the code in the likelihood expression by a Gaussian process emulator (GPE), constructed from a training set of simulations run over a set of input locations, the so-called design of experiments. This paper proposes two new algorithms for building sequential designs aiming at reducing the calibration error induced by the uncertainty of the

GPE. Although it was already mentioned by Kennedy and O'Hagan (2001) as an important research axis, few papers deal with that issue. To the best of our knowledge, Kumar (2008) has already proposed some empirical criteria for sequentially selecting the code runs. Pratola et al. (2013) have also proposed adaptive strategies whereby the Expected Improvement (EI) criterion is computed over a likelihood ratio. The EI criterion is used to optimize black box functions when a limited number of simulations is allocated (Jones et al., 1998). Recently, it has been applied to solving a problem of optimization under uncertainty when the code inputs are $(\mathbf{x}, \mathbf{u})$, where $\mathbf{x}$ denotes a vector of control variables and $\mathbf{u}$ denotes a vector of random variables (Janusevskis and Le Riche, 2013). In the same spirit, we propose new algorithms which consist in applying the EI criterion to the sum of squares of the residuals between the code outputs and the field measurements when the code inputs are $(\mathbf{x}, \boldsymbol{\tau})$ where $\boldsymbol{\tau}$ is a vector of parameters. In this way, we aim at reducing the uncertainty due to the GPE in regions of high posterior density.

This paper is divided into five sections. In Section 4.2, the statistical framework is introduced and the main features of the GPE are recalled. In Section 4.3, two new algorithms for Bayesian calibration based on the Expected Improvement criterion are presented. Their performances are illustrated on two academic examples in Section 4.4. The conclusions of this paper are provided in Section 4.5.

4.2 Calibration framework

Notations and modeling Let $r(\mathbf{x}) \in \mathbb{R}$ be a physical quantity of interest where

$$\mathbf{x} = \left(x_1, \cdots, x_d\right)^{\mathrm{T}} \in \mathscr{X} \subset \mathbb{R}^d$$

is a vector of control variables. This kind of variable is measurable in the field experiments and characterizes the system. It can include both physical variables (temperature, pressure, velocity, etc.) and design variables (height, area, etc.). We suppose that a number of field experiments, say n, has been collected. In this paper, the sites of field experiments will be referred to as the matrix

$$\mathbf{X}^f = \left[\mathbf{x}_1^f, \cdots, \mathbf{x}_n^f\right]^{\mathrm{T}} \in \mathrm{M}_{n,d}(\mathbb{R})$$

and the corresponding measurements will be referred to as the vector $\mathbf{z}^f = (z_1^f, \cdots, z_n^f)^{\mathrm{T}}$. Due mainly to the measurements errors, $\mathbf{z}^f$ is not exactly equal to $r(\mathbf{X}^f) = (r(\mathbf{X}_1^f), \cdots, r(\mathbf{X}_n^f))$. Hence, for $1 \leq i \leq n$,

$$z_i^f = r(\mathbf{x}_i^f) + \epsilon_i, \tag{4.1}$$

where

$$\epsilon_i \sim \mathscr{E}_i \underset{i.i.d.}{=} \mathscr{N}(0, \lambda^2),$$

is modeled as a white noise. The variance λ^2 is assumed known because in many cases, either the precision of the measuring device is known or it can be estimated from replicates.

Let $y_{\boldsymbol{\tau}}(\mathbf{x})$ be a deterministic computer code which predicts $r(\mathbf{x})$ where $\boldsymbol{\tau} = (\tau_1, \cdots, \tau_p)^{\mathrm{T}} \in \mathscr{T} \subset \mathbb{R}^p$ is a vector of parameters including either factors attached to the field (chemical rate, friction coefficient, etc.) or mathematical tuning factors such as a discretization step having no counterpart in physics, or perhaps both (Gang et al., 2009). The computer code is seen as a black box function, which supposes nothing is known about the connection

between the inputs $(\mathbf{x}, \boldsymbol{\tau})$ and the output $y_{\boldsymbol{\tau}}(\mathbf{x})$. A simulation refers to a code output $y_{\boldsymbol{\tau}}(\mathbf{x})$. A numerical design of experiments means a set of input locations where the code is run (Koehler and Owen, 1996). Following Kennedy and O'Hagan (2001), the computer code should be considered as an imperfect representation of the phenomenon r. Hence,

$$r(\mathbf{x}) = y_{\boldsymbol{\theta}}(\mathbf{x}) + b(\mathbf{x}), \tag{4.2}$$

where $b(\mathbf{x})$ is the code discrepancy and $\boldsymbol{\theta}$ is the "optimal" value of parameters in a certain sense. Combining (4.2) and (4.1), the statistical model which links the simulations to the field measurements is written as

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + b(\mathbf{x}_i^f) + \epsilon_i. \tag{4.3}$$

The computer code is said to be calibrated when an estimator $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ has been calculated as being the "best-fitting" parameter according to the statistical model (4.3). Due to potential non-identifiability between $\boldsymbol{\theta}$ and $b(\mathbf{x})$ in this model, the estimation of $\boldsymbol{\theta}$ requires to a prior hypothesis on the shape of $b(\mathbf{x})$. This issue has been widely studied over the past decade. Higdon et al. (2004), Kennedy and O'Hagan (2001) and many others have modeled $b(\mathbf{x})$ by a Gaussian process. Joseph and Melkote (2009) have proposed to use a more common regression model instead.

In our statistical setting, we suppose that $b(\mathbf{x})$ is negligible in the sense that it cannot be distinguished from the noise ϵ. If $b(\mathbf{x})$ is significant, then the method developed in this paper could still be applied if it has been elicited from prior expertise.

The unbiased model Let us suppose that $b(\mathbf{x}) = 0$. In other words, for at least one value in $\mathcal{T}$ denoted by $\boldsymbol{\theta}$, the computer code is supposed to be a perfect representation of the physical phenomenon r, which means that

$$\exists \boldsymbol{\theta} \in \mathcal{T} \ ; \ \forall \mathbf{x} \in \mathcal{X}, \ r(\mathbf{x}) = y_{\boldsymbol{\theta}}(\mathbf{x}). \tag{4.4}$$

Combining (4.4) and (4.1) leads to the equation

$$z_i^f = y_{\boldsymbol{\theta}}(\mathbf{x}_i^f) + \epsilon_i. \tag{4.5}$$

We have chosen to conduct a Bayesian inference of $\boldsymbol{\theta}$ because it has been shown better suited than the standard MLE [1], where flat likelihood may need regularization, especially if the dimension of $\boldsymbol{\theta}$ is high (Kumar, 2008). Let $y_{\boldsymbol{\theta}}(\mathbf{X}^f) := \left(y_{\boldsymbol{\theta}}(\mathbf{X}_1^f), \cdots, y_{\boldsymbol{\theta}}(\mathbf{X}_n^f)\right)^{\mathrm{T}}$ be the vector of code outputs running over the input field data $\mathbf{X}^f$. The posterior distribution of $\boldsymbol{\theta}$ given by the Bayes formula is a normalized version of the following product:

$$\begin{aligned} \pi(\boldsymbol{\theta}|\mathbf{z}^f) &\propto \mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta})\pi(\boldsymbol{\theta}), \\ &\propto \frac{1}{(\sqrt{2\pi}\lambda)^n} \exp\left[-\frac{1}{2\lambda^2}\mathrm{SS}(\boldsymbol{\theta})\right]\pi(\boldsymbol{\theta}), \end{aligned} \tag{4.6}$$

where

$$\mathrm{SS}(\boldsymbol{\theta}) = ||\mathbf{z}^f - y_{\boldsymbol{\theta}}(\mathbf{X}^f)||^2 \tag{4.7}$$

is the sum of squares of the residuals between the simulations $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$ and the field measurements. Throughout this paper, $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$ refers to the target posterior distribution. No closed-form expression exists for $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$ because y is usually highly non linear with respect to $\boldsymbol{\theta}$. In such cases, $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$ needs to be sampled by running an MCMC algorithm

1. Maximum Likelihood Estimation

which converges to $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$ over a very large number of samples, often several thousands (Robert and Casella, 1998). In our framework, the MCMC algorithms are thus unfeasible since each sample requires a time-consuming simulation. A way to circumvent this issue consists in setting a Gaussian Process Emulator (GPE), denoted by Y(.), as a prior distribution on $y(.)$, where (.) corresponds to a pair of inputs $(\mathbf{x}, \boldsymbol{\tau})$. In the following, we need to employ the notation $\boldsymbol{\tau} \in \mathcal{T}$ which refers to any value of the code parameter whereas $\boldsymbol{\tau} = \boldsymbol{\theta}$ refers to the optimal value to be estimated.

Gaussian process emulator The Gaussian process was first introduced within the field of computer experiments by Sacks et al. (1989). It is the most familiar surrogate model used to mimic a costly computer code. From a Bayesian perspective, the Gaussian process, denoted in this paper by Y(.), should be considered as a prior structure on the code (Currin et al., 1991):

$$\mathrm{Y}(\cdot) \sim \mathcal{GP}(m_{\boldsymbol{\beta}}(.), \sigma^2 \mathrm{C}_{\boldsymbol{\Psi}}(.,.)), \tag{4.8}$$

where

— $m_{\boldsymbol{\beta}}(.) = h(.)^{\mathrm{T}}\boldsymbol{\beta}$ where $h(.) = (h_1(.), \cdots, h_s(.))^{\mathrm{T}}$ is a vector of regression functions and $\boldsymbol{\beta} \in \mathbb{R}^s$ is a vector of location parameters,
— σ^2 the variance of the process,
— $\mathrm{C}_{\boldsymbol{\Psi}}(.,.)$ is the correlation function where $\boldsymbol{\Psi}$ is a vector of hyper-parameters including a range parameter and possibly a smoothness parameter.

The choice of the correlation function $\mathrm{C}_{\boldsymbol{\Psi}}(.,.)$ should depend on the prior information about the shape of the code output over $\mathcal{X} \times \mathcal{T}$. In addition, for both practical and theoretical reasons, a stationary function is almost always specified (Stein, 1999). Let $\mathbf{D}_{\mathrm{M}} \in (\mathcal{X} \times \mathcal{T})^{\mathrm{M}}$ be a numerical design of experiments:

$$\mathbf{D}_{\mathrm{M}} := \left[(\mathbf{x}_1^{\mathrm{D}}, \boldsymbol{\tau}_1^{\mathrm{D}}), \cdots, (\mathbf{x}_{\mathrm{M}}^{\mathrm{D}}, \boldsymbol{\tau}_{\mathrm{M}}^{\mathrm{D}})\right]^{\mathrm{T}} \in \mathrm{M}_{\mathrm{M},d+p}(\mathbb{R}). \tag{4.9}$$

After running the code over $\mathbf{D}_{\mathrm{M}}$, M simulations can be collected:

$$y(\mathbf{D}_{\mathrm{M}}) := \left(y(\mathbf{x}_1^{\mathrm{D}}, \boldsymbol{\tau}_1^{\mathrm{D}}) := y_{\boldsymbol{\tau}_1^{\mathrm{D}}}(\mathbf{x}_1^{\mathrm{D}}), \cdots, y(\mathbf{x}_{\mathrm{M}}^{\mathrm{D}}, \boldsymbol{\tau}_{\mathrm{M}}^{\mathrm{D}}) := y_{\boldsymbol{\tau}_{\mathrm{M}}^{\mathrm{D}}}(\mathbf{x}_{\mathrm{M}}^{\mathrm{D}})\right)^{\mathrm{T}}. \tag{4.10}$$

Let $\mathbf{v}_{pred}$ and $\mathbf{v}'_{pred}$ be two vectors in $\mathcal{X} \times \mathcal{T}$. Then,

— $\Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}}) = \mathrm{C}_{\boldsymbol{\Psi}}((\mathbf{x}_i^{\mathrm{D}}, \boldsymbol{\tau}_i^{\mathrm{D}}), (\mathbf{x}_j^{\mathrm{D}}, \boldsymbol{\tau}_j^{\mathrm{D}}))_{1 \leq i,j \leq \mathrm{M}}$ is the matrix of correlations between the simulations $y(\mathbf{D}_{\mathrm{M}})$,
— $\Sigma_{\boldsymbol{\Psi}}(\mathbf{v}_{pred}, \mathbf{D}_{\mathrm{M}}) = (\mathrm{C}_{\boldsymbol{\Psi}}(\mathbf{v}_{pred}, (\mathbf{x}_i^{\mathrm{D}}, \boldsymbol{\tau}_i^{\mathrm{D}}))_{1 \leq i \leq \mathrm{M}}$ is the vector of correlations between $\mathrm{Y}(\mathbf{v}_{pred})$ and each of $y(\mathbf{D}_{\mathrm{M}})$.

By conditioning the Gaussian process (4.8) on the training set of simulations $y(\mathbf{D}_{\mathrm{M}})$, the resulting process is still a Gaussian process:

$$\mathrm{Y}_{\mathrm{M}}(.) := \mathrm{Y}(.)|y(\mathbf{D}_{\mathrm{M}}) \sim \mathcal{GP}(\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(.), \mathrm{V}^{\mathrm{M}}_{\boldsymbol{\Psi},\sigma^2}(\cdot,\cdot)), \tag{4.11}$$

with the standard expressions for the conditional mean and covariance:

$$\begin{aligned}\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{v}_{pred}) =& \mathbb{E}[\mathrm{Y}_{\mathrm{M}}(\mathbf{v}_{pred})] \\ =& m_{\boldsymbol{\beta}}(\mathbf{v}_{pred}) + \Sigma_{\boldsymbol{\Psi}}(\mathbf{v}_{pred}, \mathbf{D}_{\mathrm{M}})^{\mathrm{T}} \Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}})^{-1} \left[y(\mathbf{D}_{\mathrm{M}}) - m_{\boldsymbol{\beta}}(\mathbf{v}_{pred})\right],\end{aligned} \tag{4.12}$$

and

$$\begin{aligned} \mathrm{V}^{\mathrm{M}}_{\boldsymbol{\Psi},\sigma^2}(\mathbf{v}_{pred},\mathbf{v}'_{pred}) &= \mathrm{Cov}(\mathrm{Y}_{\mathrm{M}}(\mathbf{v}_{pred}),\mathrm{Y}_{\mathrm{M}}(\mathbf{v}'_{pred})) \\ &= \sigma^2\Big(\mathrm{C}_{\boldsymbol{\Psi}}(\mathbf{v}_{pred},\mathbf{v}'_{pred}) - \Sigma_{\boldsymbol{\Psi}}(\mathbf{v}_{pred},\mathbf{D}_{\mathrm{M}})^{\mathrm{T}}\Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}})^{-1}\Sigma_{\boldsymbol{\Psi}}(\mathbf{v}'_{pred},\mathbf{D}_{\mathrm{M}})\Big). \end{aligned} \tag{4.13}$$

The GPE is given by the conditional process (4.11) which yields a stochastic prediction of the code for any input $\mathbf{v}_{pred}$ of the input space $\mathscr{X}\times\mathscr{T}$. In the case where $\mathbf{v}_{pred}$ belongs to $\mathbf{D}_{\mathrm{M}}$, the GPE interpolates the simulations $y(\mathbf{D}_{\mathrm{M}})$, that is for $1\leq i\leq \mathrm{M}$

$$\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}^{\mathrm{D}}_i,\boldsymbol{\tau}^{\mathrm{D}}_i) = y(\mathbf{x}^{\mathrm{D}}_i,\boldsymbol{\tau}^{\mathrm{D}}_i),$$

and

$$\mathrm{V}^{\mathrm{M}}_{\boldsymbol{\Psi},\sigma^2}((\mathbf{x}^{\mathrm{D}}_i,\boldsymbol{\tau}^{\mathrm{D}}_i),(\mathbf{x}^{\mathrm{D}}_i,\boldsymbol{\tau}^{\mathrm{D}}_i)) = 0$$

which is expected for such an emulator of a deterministic computer code. Lastly, the capability of a GPE to well predict the code should be checked thanks to some validation criteria (Bastos and O'Hagan, 2008). For more details about the GPE, refer to Fang et al. (2006), Rasmussen and Williams (2006), Santner et al. (2003). Other more theoretical references dedicated to asymptotic properties are Stein (1999) and Bachoc (2014).

Calibration using a GPE In Equation (4.7), the simulations $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$ are replaced with a GPE constructed from a design of experiments $\mathbf{D}_{\mathrm{M}}$. Let,

- $m_{\boldsymbol{\beta}}(\mathbf{D}_{\mathrm{M}}) = \big(h(\mathbf{x}^{\mathrm{D}}_1,\boldsymbol{\tau}^{\mathrm{D}}_1)^{\mathrm{T}}\boldsymbol{\beta},\cdots,h(\mathbf{x}^{\mathrm{D}}_{\mathrm{M}},\boldsymbol{\tau}^{\mathrm{D}}_{\mathrm{M}})^{\mathrm{T}}\boldsymbol{\beta}\big)^{\mathrm{T}}$ be the mean vector of the Gaussian process evaluated in each location of $\mathbf{D}_{\mathrm{M}}$,
- $m_{\boldsymbol{\beta}}(\mathbf{D}_{\boldsymbol{\theta}})$ and $\Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}})$ be the mean vector and the correlation matrix of the Gaussian process, each evaluated in $\mathbf{D}_{\boldsymbol{\theta}} := \big[(\mathbf{x}^f_1,\boldsymbol{\theta}),\cdots,(\mathbf{x}^f_n,\boldsymbol{\theta})\big]^{\mathrm{T}} \in \mathrm{M}_{n,d+p}(\mathbb{R})$,
- $\Sigma_{\boldsymbol{\Psi}}\big(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}\big)$ be the correlation matrix between $\mathbf{D}_{\mathrm{M}}$ and $\mathbf{D}_{\boldsymbol{\theta}}$.

Then, we now consider the available data $\mathbf{d} := (y(\mathbf{D}_{\mathrm{M}}),\mathbf{z}^f)$. The joint likelihood of $\boldsymbol{\theta}$ and $(\boldsymbol{\beta},\sigma^2,\boldsymbol{\Psi})$ is given by

$$\begin{aligned} \mathscr{L}^{\mathrm{F}}(\mathbf{d}|\boldsymbol{\theta},\sigma^2,\boldsymbol{\beta},\boldsymbol{\Psi}) \propto \frac{|\mathrm{C}_{\boldsymbol{\Psi}}|^{-1/2}}{\sigma^{\mathrm{M}+n}}\exp -\frac{1}{2\sigma^2}\Big[\big(\mathbf{d}-(m_{\boldsymbol{\beta}}(\mathbf{D}_{\mathrm{M}}),m_{\boldsymbol{\beta}}(\mathbf{D}_{\boldsymbol{\theta}}))\big)^{\mathrm{T}} \\ \mathrm{C}^{-1}_{\boldsymbol{\Psi}}\big(\mathbf{d}-(m_{\boldsymbol{\beta}}(\mathbf{D}_{\mathrm{M}}),m_{\boldsymbol{\beta}}(\mathbf{D}_{\boldsymbol{\theta}}))\big)\Big], \end{aligned} \tag{4.14}$$

where

$$\mathrm{C}_{\boldsymbol{\Psi}} = \begin{pmatrix} \Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}}) & \Sigma_{\boldsymbol{\Psi}}\big(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}\big) \\ \Sigma_{\boldsymbol{\Psi}}\big(\mathbf{D}_{\mathrm{M}},\mathbf{D}_{\boldsymbol{\theta}}\big)^{\mathrm{T}} & \Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}) + \frac{\lambda^2}{\sigma^2}\mathbf{I}_n \end{pmatrix}.$$

In the previous paragraph about the GPE, the parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\Psi}$ have been assumed to be known. If they are not, their estimation should be conducted jointly with $\boldsymbol{\theta}$ based on the full likelihood (4.14) (Higdon et al., 2004). However, inspired from the pioneering work of Cox et al. (2001), a two-step procedure can be conducted instead. This technique, known as modularization in Liu et al. (2009), is still used in a recent work dealing with calibration (Gramacy et al., 2014). It first consists in estimating the parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\Psi}$ only thanks to the simulations $y(\mathbf{D}_{\mathrm{M}})$ by maximizing the marginal density of $y(\mathbf{D}_{\mathrm{M}})$, denoted by $\mathscr{L}^{\mathrm{M}}$:

$$\mathscr{L}^{\mathrm{M}}(y(\mathbf{D}_{\mathrm{M}})|\sigma^2,\boldsymbol{\beta},\boldsymbol{\Psi}) \propto \frac{|\Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}})|^{-1/2}}{\sigma^{\mathrm{M}}} \exp\Big\{-\frac{1}{2\sigma^2}\Big[(y(\mathbf{D}_{\mathrm{M}})-m_{\boldsymbol{\beta}}(\mathbf{D}_{\mathrm{M}}))^{\mathrm{T}} \Sigma_{\boldsymbol{\Psi}}(\mathbf{D}_{\mathrm{M}})^{-1}(y(\mathbf{D}_{\mathrm{M}})-m_{\boldsymbol{\beta}}(\mathbf{D}_{\mathrm{M}}))^{\mathrm{T}}\Big]\Big\}, \quad (4.15)$$

Then, the $\mathscr{L}^{\mathrm{M}}$'s maximum likelihood estimates (MLEs) $(\hat{\sigma}^2,\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}})$ of $(\sigma^2,\boldsymbol{\beta},\boldsymbol{\Psi})$ are plugged into the likelihood of $\mathbf{z}^f$ conditional to the simulations $y(\mathbf{D}_{\mathrm{M}})$, denoted below by $\mathscr{L}^{\mathrm{C}}$:

$$\mathscr{L}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}})) \propto |\mathrm{V}^{\mathrm{M}}_{\hat{\boldsymbol{\Psi}},\hat{\sigma}^2}(\boldsymbol{\theta})+\lambda^2\mathbf{I}_n|^{-1/2}\exp\Big\{-\frac{1}{2}\Big[(\mathbf{z}^f-\mu^{\mathrm{M}}_{\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}}}(\mathbf{D}_{\boldsymbol{\theta}}))^{\mathrm{T}}) (\mathrm{V}^{\mathrm{M}}_{\hat{\boldsymbol{\Psi}},\hat{\sigma}^2}(\boldsymbol{\theta})+\lambda^2\mathbf{I}_n)^{-1}(\mathbf{z}^f-\mu^{\mathrm{M}}_{\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}}}(\mathbf{D}_{\boldsymbol{\theta}}))\Big]\Big\} \quad (4.16)$$

where

$$\mu^{\mathrm{M}}_{\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}}}(\mathbf{D}_{\boldsymbol{\theta}}) := (\mu^{\mathrm{M}}_{\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}}}(\mathbf{x}^f_1,\boldsymbol{\theta}),\cdots,\mu^{\mathrm{M}}_{\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\Psi}}}(\mathbf{x}^f_n,\boldsymbol{\theta}))^{\mathrm{T}},$$

and

$$\mathrm{V}^{\mathrm{M}}_{\hat{\boldsymbol{\Psi}},\hat{\sigma}^2}(\boldsymbol{\theta})(i,j) = \mathrm{Cov}\big(\mathrm{Y}_{\mathrm{M}}(\mathbf{x}^f_i,\boldsymbol{\theta}),\mathrm{Y}_{\mathrm{M}}(\mathbf{x}^f_j,\boldsymbol{\theta})\big).$$

This modular approach goes hand-in-hand with the intuition that the estimation of the calibration parameter $\boldsymbol{\theta}$ should be separated from the estimation of σ^2, $\boldsymbol{\beta}$ and $\boldsymbol{\Psi}$ because they are of different natures. Kennedy and O'Hagan (2001) also argue it is not a great loss to estimate σ^2, $\boldsymbol{\beta}$ and $\boldsymbol{\Psi}$ only thanks to $y(\mathbf{D}_{\mathrm{M}})$ because the number of field measurements n is usually much smaller than M. Liu et al. (2009) has put in light a poorer mixing in the MCMC routine based on the full likelihood (4.14) than based on (4.16). From now on, the conditional likelihood (4.16) is referred to as the surrogate likelihood. Let π^{C} denote the surrogate posterior distribution induced by (4.16). Then,

$$\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f,y(\mathbf{D}_{\mathrm{M}})) \propto \mathscr{L}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))\pi(\boldsymbol{\theta}). \quad (4.17)$$

The surrogate posterior (4.17) and the target posterior (4.6) are different in that $y_{\boldsymbol{\theta}}(\mathbf{X}^f)$ is replaced by the mean vector of the GPE $\mu^{\mathrm{M}}_{\boldsymbol{\beta}}(\mathbf{D}_{\boldsymbol{\theta}})$ and the conditional covariance matrix $\mathrm{V}^{\mathrm{M}}_{\hat{\boldsymbol{\Psi}},\hat{\sigma}^2}(\boldsymbol{\theta})$ is added up to $\lambda^2\mathbf{I}_n$. Contrary to the target posterior (4.6), the surrogate posterior is cheap to evaluate, enabling to perform an MCMC algorithm in order to estimate $\boldsymbol{\theta}$.

Remark The first stage of the modular approach neglects the uncertainty of the parameters σ^2, $\boldsymbol{\beta}$ and $\boldsymbol{\Psi}$ by fixing them to their MLE. However, a possible manner to take into account their uncertainty would consist in adopting a Bayesian inference of σ^2, $\boldsymbol{\beta}$ and $\boldsymbol{\Psi}$ in the same way than $\boldsymbol{\theta}$. For instance, if a Jeffreys prior distribution is specified on $(\boldsymbol{\beta},\sigma^2)$, then the distribution of the GPE follows a Student distribution (Santner et al., 2003). Yet unfortunately, the conditional likelihood (4.16) has no closed-form expression anymore, causing an additional issue which is beyond the scope of this paper.

In presence of a code discrepancy $b(\mathbf{x})$ The calibration setting (4.6) has to be modified to prevent overfitting of $\boldsymbol{\theta}$, as showed in Bayarri et al. (2007). In Higdon et al. (2004) and Bayarri et al. (2007), $b(\mathbf{x})$ is modeled by a zero mean Gaussian process

$$b(.) \sim \mathscr{GP}(0,\sigma_b^2\mathrm{C}_{\boldsymbol{\Psi}_b}(.,.))$$

where (.) corresponds here to an input $\mathbf{x}$. In this case, the logarithm of the surrogate likelihood (4.16) is now proportional to a weighted sum of squares of the residuals:

$$\mathrm{SS}(\boldsymbol{\theta}) = (\mathbf{z}^f - y_{\boldsymbol{\theta}}(\mathbf{X}^f))^{\mathrm{T}}(\mathrm{V}_b^f + \lambda^2 \mathbf{I}_n)^{-1}(\mathbf{z}^f - y_{\boldsymbol{\theta}}(\mathbf{X}^f)) \tag{4.18}$$

where $\mathrm{V}_b^f(i,j) = \sigma_b^2 \mathrm{C}_{\boldsymbol{\Psi}_b}(\mathbf{x}_i^f, \mathbf{x}_j^f)$. Sometimes, the physical context helps us fix both σ_b^2 and $\boldsymbol{\Psi}_b$ to plausible values, as done in Craig et al. (2001). In such cases, V_b^f becomes known and the algorithms that we present in Section 4.3 will still be practicable based on (4.18) instead of (4.7).

Main goal of the paper Our work focuses on reducing the distance between the surrogate posterior distribution (4.17) and the target posterior distribution (4.6). The Kullback-Leibler (KL) divergence shows interesting theoretical properties to measure how far a probability distribution is from a reference one (Cover and Thomas, 1991). It is written as

$$\mathrm{KL}\Big(\pi(\boldsymbol{\theta}|\mathbf{z}^f)||\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))\Big) = \int_{\mathscr{T}} \pi(\boldsymbol{\theta}|\mathbf{z}^f)\Big(\log(\pi(\boldsymbol{\theta}|\mathbf{z}^f)) - \log(\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))\Big)\mathrm{d}\boldsymbol{\theta}. \tag{4.19}$$

The design of experiments $\mathbf{D}_{\mathrm{M}}$ By using results of approximation theory, we can prove the proposition below.

Proposition 4.2.0.1 *Under the following assumptions:*

- $\pi(\boldsymbol{\theta})$ *has a bounded support* $\mathscr{T}$,
- *the code output* $y_{\boldsymbol{\tau}}(\mathbf{x})$ *is uniformly bounded on* $\mathscr{T} \times \mathscr{X}$,
- *the correlation function (kernel) is a classical radial basis function (Schaback, 1995),*
- y *lies in the associated Reproducing Kernel Hilbert Space,*
- *the covering distances associated with the sequence of designs* $(\mathbf{D}_{\mathrm{M}})_{\mathrm{M}}$ *tends to* 0 *with* $\mathrm{M} \to \infty$ *(see Appendix),*

then, we have:

$$\lim_{\mathrm{M}\to\infty} KL\Big(\pi(\boldsymbol{\theta}|\mathbf{z}^f)||\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))\Big) = 0. \tag{4.20}$$

Proof. (4.20) results from the uniform convergence of $|\log(\pi(\boldsymbol{\theta}|\mathbf{z}^f)) - \log(\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))|$ to 0 over $\mathscr{T}$ when M tends to ∞ (see Appendix). Then, we can exchange limit and integral in (4.19), which completes the proof.

This proposition reflects the fact that the calibration based on the surrogate posterior (4.17) is consistent, in other words the larger $\mathbf{D}_{\mathrm{M}}$ the closer the surrogate posterior (4.17) to the target posterior (4.6). However, when M is small with respect to the dimension of the input space $\mathscr{X} \times \mathscr{T}$, (4.17) can be significantly different from (4.6) leading to a large KL divergence (4.19). Although an approximate posterior distribution constructed from an accurate GPE is likely to yield a small value of the KL divergence (4.19), our own experience has shown that such behavior is not systematic.

In practical use, $\mathbf{D}_{\mathrm{M}}$ is built as a space-filling design on the input space $\mathscr{X} \times \mathscr{T}$ (Pronzato and Muller, 2012), and hence does not take that $\mathbf{x}$ and $\boldsymbol{\tau}$ have different roles to play in calibration. In fact, our interest is not to predict well the computer code over $\mathscr{X} \times \mathscr{T}$ but rather to minimize the KL divergence (4.19), which can be developed as

$$\begin{aligned}\mathrm{KL}(\pi(\boldsymbol{\theta}|\mathbf{z}^f)||\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}})) = \int_{\mathscr{T}} \pi(\boldsymbol{\theta}|\mathbf{z}^f)(\mathrm{C} - \mathrm{C}_{\mathrm{M}}(\boldsymbol{\theta}))\mathrm{d}\boldsymbol{\theta} + \frac{1}{2}\int_{\mathscr{T}} \pi(\boldsymbol{\theta}|\mathbf{z}^f) \\ \Big((\mathbf{z}^f - \mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))^{\mathrm{T}}(\mathrm{V}^{\mathrm{M}}_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n)^{-1}(\mathbf{z}^f - \mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}})) - \mathrm{SS}(\boldsymbol{\theta})\Big)\mathrm{d}\boldsymbol{\theta}\end{aligned} \tag{4.21}$$

where

$$C = -\frac{n}{2}\log\lambda^2, \tag{4.22}$$

and

$$C_M(\boldsymbol{\theta}) = -\frac{1}{2}\log|V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n|. \tag{4.23}$$

According to Equation (4.21), the KL divergence could be minimized by reducing the uncertainty of the GPE over input locations in the subspace $\mathbf{X}^f \times \mathcal{T}^n \subset (\mathcal{X}\times\mathcal{T})^n$ and above all where the target posterior distribution $\pi(\mathbf{z}^f|\boldsymbol{\theta})$ is high. In Section 4.3, we propose new algorithms for building proper numerical designs $\mathbf{D}_M$ designed for this purpose.

4.3 Adaptive designs for calibration

To identify the global minimum of a costly black box code, denoted by f (to avoid confusion with the calibration setting), Expected Improvement (EI) criterion-based strategies can be performed (Jones et al., 1998). They consist in identifying sequentially the input locations where the code f should be run to be close to the global minimum, which is relevant when only a small number of simulations is allocated. Assuming k simulations $f(\mathbf{D}_k)$ have already been run, the EI criterion assesses the expected improvement of a new run (numbered $k+1$) in terms of getting close to the unknown global minimum of f. Let $\mathbf{v}_{k+1}$ be the input where the EI value is at its highest, meaning that,

$$\begin{aligned}\mathbf{v}_{k+1} &= \operatorname*{argmax}_{\mathbf{v}} EI_k(\mathbf{v}),\\ &= \operatorname*{argmax}_{\mathbf{v}} \mathbb{E}[(m_k - F_k(\mathbf{v}))\mathbf{1}_{F_k(\mathbf{v})<m_k}],\end{aligned} \tag{4.24}$$

where

— $F_k(\mathbf{v})$ is the current GPE which is built from $\mathbf{D}_k$,

— $m_k = \min\{f(\mathbf{v}_1),\cdots,f(\mathbf{v}_{k-1}),f(\mathbf{v}_k)\}$ is the current value for the minimum.

If a deterministic emulator were used instead of $F_k(\mathbf{v})$, for instance the mean $\mu(\mathbf{v})$ of $F_k(\mathbf{v})$, the EI criterion would be just the difference $m_k - \mu(\mathbf{v})$ if $\mu(\mathbf{v}) < m_k$ and 0 if $\mu(\mathbf{v}) > m_k$. Given that $F_k(\mathbf{v})$ is stochastic, Equation (4.24) is written as the expectation of this truncated difference with respect to the distribution of F_k. The algorithm that consists of running the code at the input $\mathbf{v}_{k+1}$ then updating the emulator and starting again is called Efficient Global Optimization (EGO) (Jones et al., 1998). The convergence of the EGO algorithm to the global minimum of f has been proven with respect to some assumptions about both the smoothness of the code and the correlation function of the GPE (Vazquez and Bect, 2010). In its current use, the algorithm is stopped when the number of allocated simulations is over or when the improvement of m_k becomes negligible. According to this last criterion, EGO requires less simulations than other optimization methods with comparable levels of performance (Ginsbourger, 2009).

EI designed for calibration Our contribution now consists in resorting to the EI criterion for the sum of squares of the residuals function $SS(\boldsymbol{\theta})$:

$$EI_k(\boldsymbol{\theta}) = \mathbb{E}\left[\left(m_k - SS_k(\boldsymbol{\theta})\right)\mathbf{1}_{SS_k(\boldsymbol{\theta})\leq m_k}\right] \in [0, m_k], \tag{4.25}$$

where

— $m_k := \min\{SS(\boldsymbol{\theta}_1),\cdots,SS(\boldsymbol{\theta}_{k-1}),SS(\boldsymbol{\theta}_k)\}$ and $SS(\cdot)$ denotes the sum of squares computed from actual runs of the computer code y,

— $\mathrm{SS}_k(.)$ denotes the sum of squares of the residuals where y is replaced with the GPE conditional to $y(\mathbf{D}_k)$, denoted by

$$\mathrm{Y}_k(.) := \mathrm{Y}(.)|y(\mathbf{D}_k),$$

where the subscript k now refers to the current iteration of the algorithm. $\mathrm{SS}_k(.)$ is thus a random process and its distribution inherits from the current GPE. At step k, nk new simulations need to be run to update m_k. Hence, the design $\mathbf{D}_k$ contains all the simulations $y_{\boldsymbol{\theta}_j}(\mathbf{x}_i^f)$ for all $1 \leq i \leq n$ and $1 \leq j \leq k$ (should not be mistaken for the notation in Section 4.2 where M has referred to the number of simulations). Let $\boldsymbol{\theta}^\star$ be the maximum of (4.25):

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathrm{EI}_k(\boldsymbol{\theta}).$$

Justification for minimizing $\mathrm{SS}(\boldsymbol{\theta})$ As explained at the end of the previous section, running the code over input locations in $\mathbf{X}^f \times \mathcal{T}$ where $\pi(\mathbf{z}^f|\boldsymbol{\theta})$ is high reduces the distance between $\pi^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta}, y(\mathbf{D}_{\mathrm{M}}))$ and $\pi(\mathbf{z}^f|\boldsymbol{\theta})$. In addition, when no expert knowledge is available on the value of $\boldsymbol{\theta}$ (except perhaps a lower and an upper bound), a noninformative prior should be specified for $\boldsymbol{\theta}$ (Box and Tiao, 1973; Kass and Wasserman, 1996). In such cases, the higher the target posterior distribution $\pi(\mathbf{z}^f|\boldsymbol{\theta})$, the lower $\mathrm{SS}(\boldsymbol{\theta})$. The KL divergence (4.19) can be then reduced faster by running the code over the input locations $(\mathbf{x}_i^f, \boldsymbol{\theta}^\star) \in \mathbf{X}^f \times \mathcal{T}$ where $\mathrm{SS}(\boldsymbol{\theta}^\star)$ is small. Such locations $\boldsymbol{\theta}^\star$ can be identified by maximizing the EI criterion.

Remark In some works, calibration consists in minimizing a gap between the code outputs and the available field measurements, such as $\mathrm{SS}(\boldsymbol{\theta})$ (Wong et al., 2014) or a weighted sum of squares like (4.18) (Joseph and Melkote, 2009). For our part, the EI criterion applied to $\mathrm{SS}(\boldsymbol{\theta})$ is just a circuitous way to efficiently build $\mathbf{D}_{\mathrm{M}}$ for Bayesian calibration.

EGO algorithms Algorithm 1 corresponds to an exact EGO algorithm based on Equation (4.25). It aims at identifying the input locations $(\mathbf{x}_i^f, \boldsymbol{\theta}^\star) \in \mathbf{X}^f \times \mathcal{T}$ which will be added up sequentially to an initial design $\mathbf{D}_0$ for M iterations. Algorithm 2 is a one at time algorithm and should be understood as an approximation of Algorithm 1.

Algorithm 1

Initialization

— Build a maximin Latin Hypercube Design (LHD) $\mathbf{D}_0 \subset \mathcal{X} \times \mathcal{T}$ of size M_0.

— Run the code over $\mathbf{D}_0$.

— Compute $\hat{\boldsymbol{\theta}}_1$ as the maximum of $\pi^{\mathrm{C}}(\boldsymbol{\theta}|\mathbf{z}^f, y(\mathbf{D}_0))$.

— $\mathbf{D}_1 = \mathbf{D}_0 \cup \{(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_1)\}_{1 \leq i \leq n}$.

— Update the Gaussian process distribution after running the code over $\{(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_1)\}_{1 \leq i \leq n}$.

— Set $m_1 := \mathrm{SS}(\hat{\boldsymbol{\theta}}_1)$.

From $k = 1$**, repeat the following steps as long as** $\mathrm{M}_0 + n \times (k+1) \leq \mathrm{M}$**.**

Step 1 Find an estimate $\hat{\boldsymbol{\theta}}_{k+1}$ of $\boldsymbol{\theta}_{k+1}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathrm{EI}_k(\boldsymbol{\theta})$.

Step 2 $\mathbf{D}_{k+1} = \mathbf{D}_k \cup \{(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\}_{1 \leq i \leq n}$.

Step 3 Run the code over all new locations $\{(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\}_{1 \leq i \leq n}$.

Step 4 Update the Gaussian process distribution based on $y(\mathbf{D}_{k+1})$.

Step 5 Let $m_{k+1} := \min\{m_1, \cdots, m_k, \mathrm{SS}(\hat{\boldsymbol{\theta}}_{k+1})\}$.

Because the distribution of the GPE is updated at Step 4, the hyper-parameters $\boldsymbol{\beta}$, σ^2 and $\boldsymbol{\Psi}$ are re-estimated, as done in the seminal EGO work (Jones et al., 1998). Algorithm 1 could be efficiently performed by running the new simulations at Step 3 simultaneously on several computer nodes. We present below the steps of a one-at-a-time algorithm well-suited when the computer system has a single processor.

Algorithm 2

Initialization is similar to Algorithm 1 except that $\mathbf{D}_1 = \mathbf{D}_0 \cup \{(\mathbf{x}^\star, \hat{\boldsymbol{\theta}}_1)\}$ where

$$\mathbf{x}^\star = \underset{\mathbf{x}_i^f \in \mathbf{X}^f}{\operatorname{argmax}} \mathrm{Crit}(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_1) \text{ (see Equations (4.26) and (4.28) below).}$$

For $k = 1, \cdots, \mathrm{M} - \mathrm{M}_0$, **repeat the same steps as in Algorithm** 1 **except that Step** 3 **is replaced with Step** $\tilde{3}$ **and Step** 5 **is replaced with Step** $\tilde{5}$**.**

Step $\tilde{3}$ Run the code in $(\mathbf{x}^\star, \hat{\boldsymbol{\theta}}_{k+1})$ where

$$\mathbf{x}^\star = \underset{\mathbf{x}_i^f \in \mathbf{X}^f}{\operatorname{argmax}} \mathrm{Crit}(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1}) \text{ (see Equations (4.26) and (4.28) below).}$$

Step $\tilde{5}$ $m_{k+1} := \min\{\mathbb{E}[\mathrm{SS}_k(\hat{\boldsymbol{\theta}}_1)], \cdots, \mathbb{E}[\mathrm{SS}_k(\hat{\boldsymbol{\theta}}_k)], \mathbb{E}[\mathrm{SS}_k(\hat{\boldsymbol{\theta}}_{k+1})]\}$

Algorithm 2 should be understood as an approximation of Algorithm 1 where only a single simulation $y_{\hat{\boldsymbol{\theta}}_{k+1}}(\mathbf{x}^\star)$ is run at each iteration. In the following, two criteria are proposed to choose $\mathbf{x}^\star$ among $\mathbf{X}^f$. As the current minimum m_k in Algorithm 1 cannot be computed anymore, we have replaced it by the minimum of the expectations $\mathbb{E}[\mathrm{SS}_k(\hat{\boldsymbol{\theta}}_i)]$ for $1 \leq i \leq k+1$ with respect to $\mathrm{Y}_k(.)$.

The first criterion is done to run the code at the input location $(\mathbf{x}^\star, \hat{\boldsymbol{\theta}}_k) \in \mathbf{X}^f \times \mathscr{T}$ where the variance of $\mathrm{Y}_k(.)$ is at the highest. Hence,

$$\mathrm{Crit}(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1}) = \mathbb{V}[\mathrm{Y}_k(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})]\,. \tag{4.26}$$

As the variance of the Gaussian process decreases on the space $\mathbf{X}^f \times \mathscr{T}$, the surrogate posterior distribution (4.17) gets closer to the target posterior distribution (4.6), which justifies (4.26). Yet, a better way might perhaps consist in aiming for a reduction of the GPE uncertainty at an input location $(\mathbf{x}^\star, \hat{\boldsymbol{\theta}}_{k+1})$ where the code $y_{\boldsymbol{\theta}}(\mathbf{x}^\star)$ is highly variable over $\boldsymbol{\theta}$, meaning that $\mathbf{x}^\star$ is important for calibration. We thus introduce a second criterion which does a trade-off between the calibration goal and (4.26). A normalized version of it is written as

$$\mathrm{Crit}(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1}) = \frac{\mathbb{V}\big(\mathrm{Y}_k(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\big)}{\max\limits_{i=1,\cdots,n} \mathbb{V}\big(\mathrm{Y}_k(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\big)} \times \frac{\mathbb{V}[y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)]}{\max\limits_{i=1,\cdots,n} \mathbb{V}[y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)]}, \tag{4.27}$$

where $\mathbb{V}[y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)]$ is taken with respect to $\pi(\boldsymbol{\theta})$. Since the code is unknown, an approximation of (4.27) can be based on the mean of $\mathrm{Y}_k(.)$:

$$\mathrm{Crit}(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1}) = \frac{\mathbb{V}\big(\mathrm{Y}_k(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\big)}{\max\limits_{i=1,\cdots,n} \mathbb{V}\big(\mathrm{Y}_k(\mathbf{x}_i^f, \hat{\boldsymbol{\theta}}_{k+1})\big)} \times \frac{\mathbb{V}[\mu^k_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}_i^f, \boldsymbol{\theta})]}{\max\limits_{i=1,\cdots,n} \mathbb{V}[\mu^k_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}_i^f, \boldsymbol{\theta})]}\,. \tag{4.28}$$

Remark A typical problem inherent to sequential designs is when two input locations come very close, making the covariance matrix numerically singular and thus difficult to invert. This issue can arise when both $\hat{\boldsymbol{\theta}}_k$ is too close to a previous iteration $\hat{\boldsymbol{\theta}}_{k'}$ and $\mathbf{x}^\star$ is almost the same at iteration k, then at $k' > k$. The usual way to circumvent this issue consists in adding a small diagonal matrix to the covariance matrix $\mathrm{V}^{k'}_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})$ of the GPE, called nugget effect.

Remark The expectation in (4.25) is taken with respect to the multivariate Gaussian distribution induced by the distribution of the GPE. Another way would have been to emulate the function $\mathrm{SS}(\boldsymbol{\theta})$ as a Gaussian process, forcing n simulations to be run at each iteration. Doing so, one at a time algorithms would have been impossible, making this strategy irrelevant for large values of n.

Computation of the criterion By expanding Equation (4.25), we have :

$$\mathrm{EI}_k(\boldsymbol{\theta}) = m_k\left[\mathbb{P}[\mathrm{SS}_k(\boldsymbol{\theta}) < m_k] - \frac{\mathbb{E}[\mathrm{SS}_k(\boldsymbol{\theta})\mathbf{1}_{\mathrm{SS}_k(\boldsymbol{\theta})\leq m_k}]}{m_k}\right] > 0\,, \tag{4.29}$$

implying

$$\mathbb{E}[\mathrm{SS}_k(\boldsymbol{\theta})\mathbf{1}_{\mathrm{SS}_k(\boldsymbol{\theta})\leq m_k}] \leq m_k\mathbb{P}[\mathrm{SS}_k(\boldsymbol{\theta}) < m_k]. \tag{4.30}$$

Except in the trivial case $n = 1$, no closed form can be calculated for (4.29). The expression of $\mathrm{EI}_k(\boldsymbol{\theta})$ is proportional to the sum of $\mathbb{P}[\mathrm{SS}_k(\boldsymbol{\theta}) < m_k]$ which is the probability of sampling inside the hypersphere $\mathrm{B}(0,\sqrt{m_k})$ from a multivariate Gaussian distribution and a second term which is the expectation of the right truncated $\mathrm{SS}_k(\boldsymbol{\theta})$ with respect to m_k. The first term can be calculated either as an infinite series in central chi-square distribution (Sheil and Muircheartaigh, 1977) or thanks to an advanced sampling rejection method (Ellis and Maitra, 2007). This second method should be preferably performed because the second term has to be estimated using MCMC. The minimization of (4.29) can be performed in a greedy fashion where $\hat{\boldsymbol{\theta}}_{k+1}$ is taken as the value which maximizes $\mathrm{EI}_k(\boldsymbol{\theta})$ over a grid $\mathrm{G} \in \mathscr{T}$. However, the computation of $\mathrm{EI}_k(\boldsymbol{\theta})$ could be avoided for some candidates of G, as explained hereafter.

Computation of $\hat{\boldsymbol{\theta}}_{k+1}$

Let $\mathrm{Y}_k(\mathbf{D}_{\boldsymbol{\theta}})$ be the random vector $\left(\mathrm{Y}_k(\mathbf{x}_1^f,\boldsymbol{\theta}),\cdots,\mathrm{Y}_k(\mathbf{x}_n^f,\boldsymbol{\theta})\right)$.

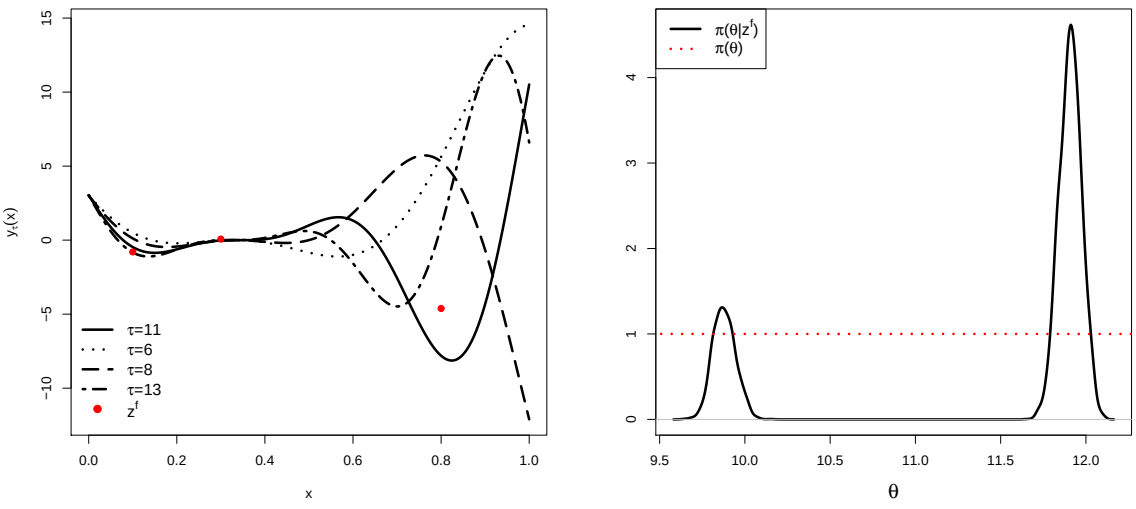
1. Compute $\mathbb{P}[\mathbf{z}^f - \mathrm{Y}_k(\mathbf{D}_{\boldsymbol{\theta}}) \in [-m_k,m_k]^n]$ which is an upper bound of $\mathbb{P}[\mathrm{SS}_k(\boldsymbol{\theta}) \leq m_k]$ for each $\boldsymbol{\theta}$ of G,
2. Let $\tilde{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\mathrm{G}}{\operatorname{argmax}}\,\mathbb{P}[\mathbf{z}^f - \mathrm{Y}_k(\mathbf{D}_{\boldsymbol{\theta}}) \in [-m_k,m_k]^n]$ be a reference value.
3. Compute $\mathrm{EI}_k(\tilde{\boldsymbol{\theta}})$,
4. Build the sub-grid $\tilde{\mathrm{G}} = \{\boldsymbol{\theta} \in \mathscr{T}\ ;\ \mathrm{EI}_k(\tilde{\boldsymbol{\theta}}) \leq \mathbb{P}[\mathbf{z}^f - \mathrm{Y}_k(\mathbf{D}_{\boldsymbol{\theta}}) \in [-m_k,m_k]^n]\} \subset \mathrm{G}$,
5. Compute $\mathrm{EI}_k(\boldsymbol{\theta})$ for the values of the sub-grid $\tilde{\mathrm{G}}$,
6. Let $\hat{\boldsymbol{\theta}} = \underset{\theta\in\tilde{\mathrm{G}}}{\operatorname{argmax}}\,\mathrm{EI}_k(\boldsymbol{\theta})$.

For $\boldsymbol{\theta} \in \mathrm{G}\setminus\tilde{\mathrm{G}}$, we have $\mathrm{EI}_k(\tilde{\boldsymbol{\theta}}) \geq \mathbb{P}[\mathbf{z}^f - \mathrm{Y}_k \in [-m_k,m_k]^n]$ implying $\mathrm{EI}_k(\tilde{\boldsymbol{\theta}}) > \mathrm{EI}_k(\boldsymbol{\theta})$. Hence, there is no need to compute $\mathrm{EI}_k(\boldsymbol{\theta})$. Unfortunately, this algorithm is only relevant when n is small because in higher dimensions, the hypercube $[-m_k,m_k]^n$ has a much larger volume than the hypersphere.

Such a greedy optimization works very well on our toy examples, especially in small dimensions of $\mathcal{T}$ because G can be constructed fine enough (see Section 4.4). In higher dimensions, the EI criterion could be maximized alternatively over several different grids as iterations of the EGO algorithm (see Section 4.4). However, if G is coarser, we can expect our algorithms stay efficient in terms of reducing the KL divergence because they do not aim at converging precisely to the global minimum of $\mathrm{SS}(\boldsymbol{\theta})$, but rather identifying the area of the input space where $\mathrm{SS}(\boldsymbol{\theta})$ is small.

4.4 Simulation study

Figure 4.1 – *Left: the function* $y_\tau(x) = (6x-2)^2 \times \sin(\tau x - 4)$ *on* $[0,1]$ *for several values of* $\tau \in [5,15]$. *Red dots are the field measurements* $(\mathbf{X}^f, \mathbf{z}^f)$ *generated by Equation (4.32). Right: the target posterior distribution (Case* 1*).*



A 2D example Let us assume that the computer code is given by the following function:

$$y_\tau : x \longrightarrow y_\tau(x) = (6x-2)^2 \times \sin(\tau x - 4), \tag{4.31}$$

where $x \in \mathcal{X} = [0,1]$ and $\tau \in \mathcal{T} = [5,15]$. For $1 \leq i \leq n$, the field data $\mathbf{z}^f$ are generated by

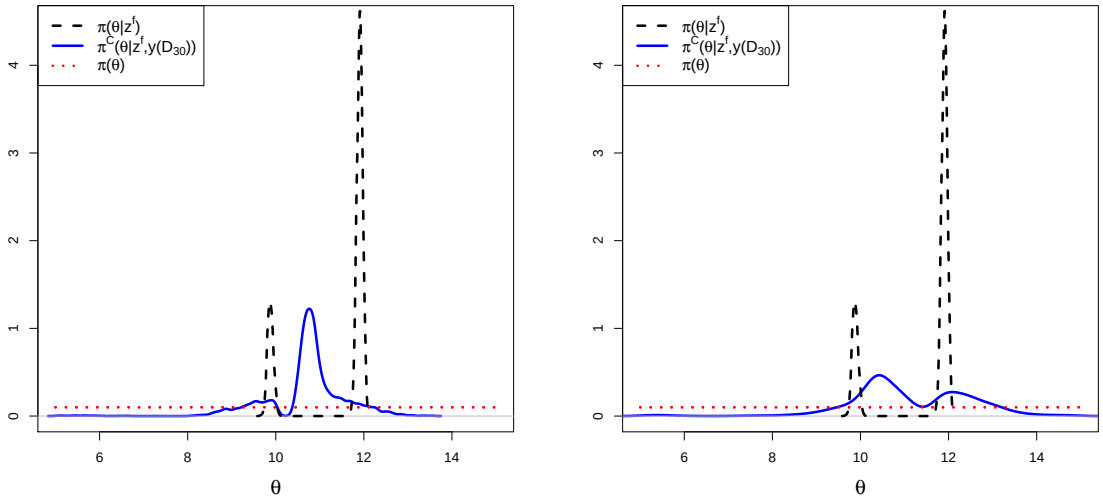
$$z_i^f = y_\theta(x_i^f) + \epsilon_i, \tag{4.32}$$

where $\theta = 12$ and $\epsilon_i \sim \mathcal{N}(0, 0.3^2)$. Bayesian calibration of the function (4.31) is done by sampling the target posterior distribution (4.6) (see Figure 4.1) where the prior distribution $\pi(\theta)$ is chosen as uniform on $[5,15]$:

$$\pi(\theta) \propto \mathbf{1}_{[5,15]}. \tag{4.33}$$

Now, Bayesian calibration of (4.31) is done by sampling the surrogate posterior distribution (4.17). Two cases are considered: with $n = 3$ field measurements, then with $n = 9$ field measurements.

Figure 4.2 – *Sampling of (4.17) from two different maximin LHD of size* M = 30 *(using the R library MCMCpack).*



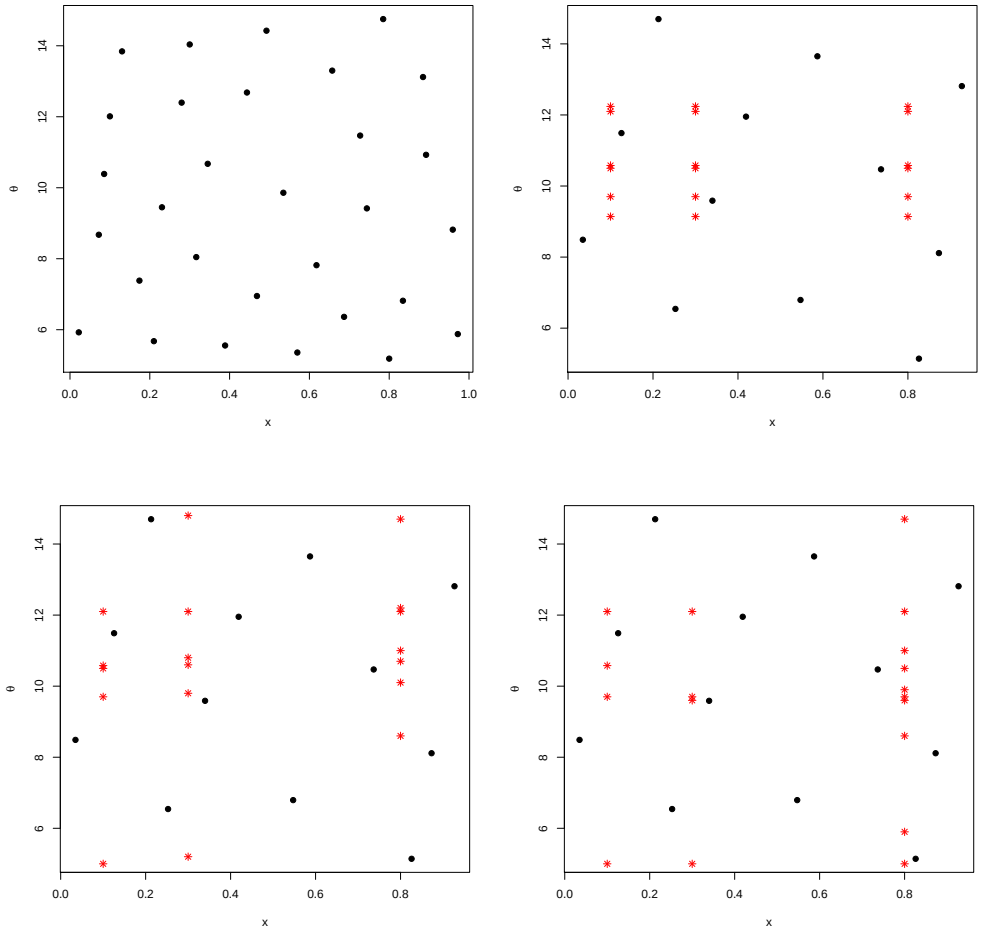
Case 1: $\mathbf{X}^f = (0.1, 0.3, 0.8)$ The surrogate posterior distribution (4.17) is sampled from a GPE with a constant mean $m_{\boldsymbol{\beta}} = m$ and a Matern 5/2 correlation function. All the parameters m, σ^2, Ψ have been estimated using the modular approach based on a maximin LHD of size M = 30 constructed with the R library DiceDesign. This calibration procedure is repeated twice by sampling (4.17) from two different maximin LHD. According to the first one, as illustrated in Figure 4.2 (right), the regions of high posterior density roughly match with those of the target posterior distribution but the two modes are reversed in terms of height. According to the second one, as illustrated in Figure 4.2 (left), the posterior distribution is very erroneous. Such calibrations where the surrogate posterior distribution is not close to the target one are unwanted, which justifies constructing some oriented designs for calibration instead of default space filling designs such as maximin LHD. In the following, three adaptive strategies building on the EGO algorithms from Section 4.3 are presented:

1. a first version based on Algorithm 1. At each iteration, the code is run at all three inputs $(0.1, \theta_k)$, $(0.3, \theta_k)$, $(0.8, \theta_k)$ where θ_k is the current parameter maximizing the EI criterion. An example of a design obtained with this algorithm is displayed in Figure 4.3 (upper right).

2. a second version based on Algorithm 2. The code is run at a single input $(x^\star, \theta_k)$ where $x^\star$ comes from the input (x_i, θ_k) having the highest variance (4.26) among the three inputs $(0.1, \theta_k)$, $(0.3, \theta_k)$, $(0.8, \theta_k)$. An example of a design obtained with this algorithm is displayed in Figure 4.3 (bottom left).

3. a third version based on Algorithm 2. The code is run at a single input $(x^\star, \theta_k)$ where $x^\star$ comes from the input (x_i, θ_k) which maximizes the criterion (4.28) among the three inputs $(0.1, \theta_k)$, $(0.3, \theta_k)$, $(0.8, \theta_k)$. An example of a design obtained with this algorithm is displayed in Figure 4.3 (bottom right)

Figure 4.3 – *Case* 1. *Upper left: a maximin LHD* ($M = 30$). *Upper right: a sequential design built from Version* 1 ($M_0 = 12$). *Bottom left: a sequential design built from Version* 2 ($M_0 = 12$). *Bottom right: a sequential design built from Version* 3 ($M_0 = 12$). *The black dots are the initial design. The red stars are the new experiments selected from the EI criterion.*



Let us now assess how good is the calibration by computing $\mathrm{KL}(\pi(\theta|\mathbf{z}^f)||\pi^{\mathrm{C}}(\theta|\mathbf{z}^f, y(\mathbf{D}_{\mathrm{M}}))$ according to the kind of design $\mathbf{D}_{\mathrm{M}}$ which is used. The proportion of the time that the 95% credibility interval of (4.17) covers the true value is computed as well. The robustness of the results is checked by repeating sampling of the surrogate posterior distribution (4.17) many times. Here, calibration is performed from 50 data sets $(\mathbf{X}^f, \mathbf{z}^f)$ and for each of them, 50 calibrations derived from 50 different designs have been conducted. In Figure 4.4, the boxplots of the KL divergence and the boxplots of the coverage are each plotted against the kind of design (including the maximin LHD and the sequential designs coming from the EGO algorithms). Each value of the boxplot is computed as the mean of the criterion over the 50 calibrations derived from a particular data set $(\mathbf{X}^f, \mathbf{z}^f)$. A sharp decrease in the KL divergence is noticed when the calibration is done with a sequential design. We can see that both versions 2 and 3 have the lowest values for this criterion. The reason is that Algorithm 2 can move more quickly inside the parameter space $\mathcal{T}$, which is expected when the target posterior distribution is multimodal. Results about coverage look the same although some lower values can be seen. In such cases, the coverage is poor because the true value ($\theta = 12$) is not covered by the 95% interval of the target posterior distribution. The same feature is thus expected with the surrogate posterior distribution.

Figure 4.4 – *Case* 1. *Left: boxplots of the KL divergence computed between the target posterior distribution and the surrogate posterior distribution (using the R library FNN). Right: ability of the* 95% *credibility interval to cover the true value* ($\theta = 12$).

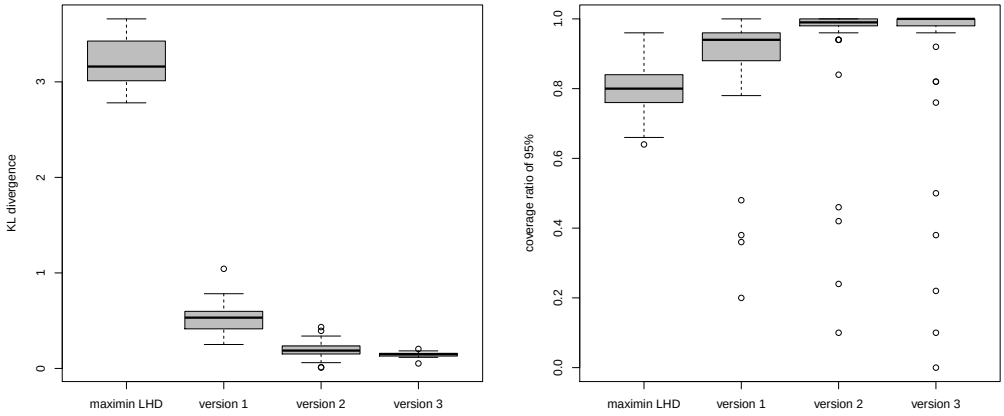
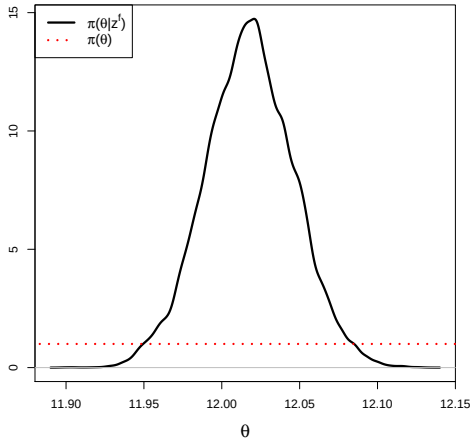


Figure 4.5 – *The target posterior distribution (Case* 2*).*



Case 2**:** $\mathbf{X}^f = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$ The field data $\mathbf{z}^f$ are still generated by

$$z_i^f = y_\theta(x_i^f) + \epsilon_i \quad \text{for} \quad i = 1, \cdots, n, \tag{4.34}$$

where $\theta = 12$ and $\epsilon_i \sim \mathcal{N}(0, 0.3^2)$. As n is larger, the target posterior distribution $\pi(\theta|\mathbf{z}^f)$ now has a single narrow mode around the true value. Similar to the first case, the calibration results are improved by using adaptive designs (see Figure 4.7). As the number of field data is larger, the one-at-a-time Versions 2 and 3 outperform Version 1 because they do not need all the evaluations corresponding to a θ around 14 to discard this area, and evaluations remain to refine the sampling around $\theta = 12$ (see Figure 4.6).

A 6**D example** Inspired from Saltelli et al. (2000), let us now assume that the computer code is given by the following function:

$$y_{\boldsymbol{\tau}} : \mathbf{x} \longrightarrow y_{\boldsymbol{\tau}}(\mathbf{x}) = \prod_{i=1}^{3} \frac{|4x_i - 2| + \tau_i}{1 + \tau_i}. \tag{4.35}$$

Figure 4.6 – *Case* 2. *Upper left: a maximin LHD* ($M = 30$). *Upper right: a sequential design built from Version* 1 ($M_0 = 12$). *Bottom left: a sequential design built from Version* 2 ($M_0 = 12$). *Bottom right: a sequential design built from Version* 3 ($M_0 = 12$). *The black dots are the initial design. The red stars are the new experiments selected from the EI criterion.*

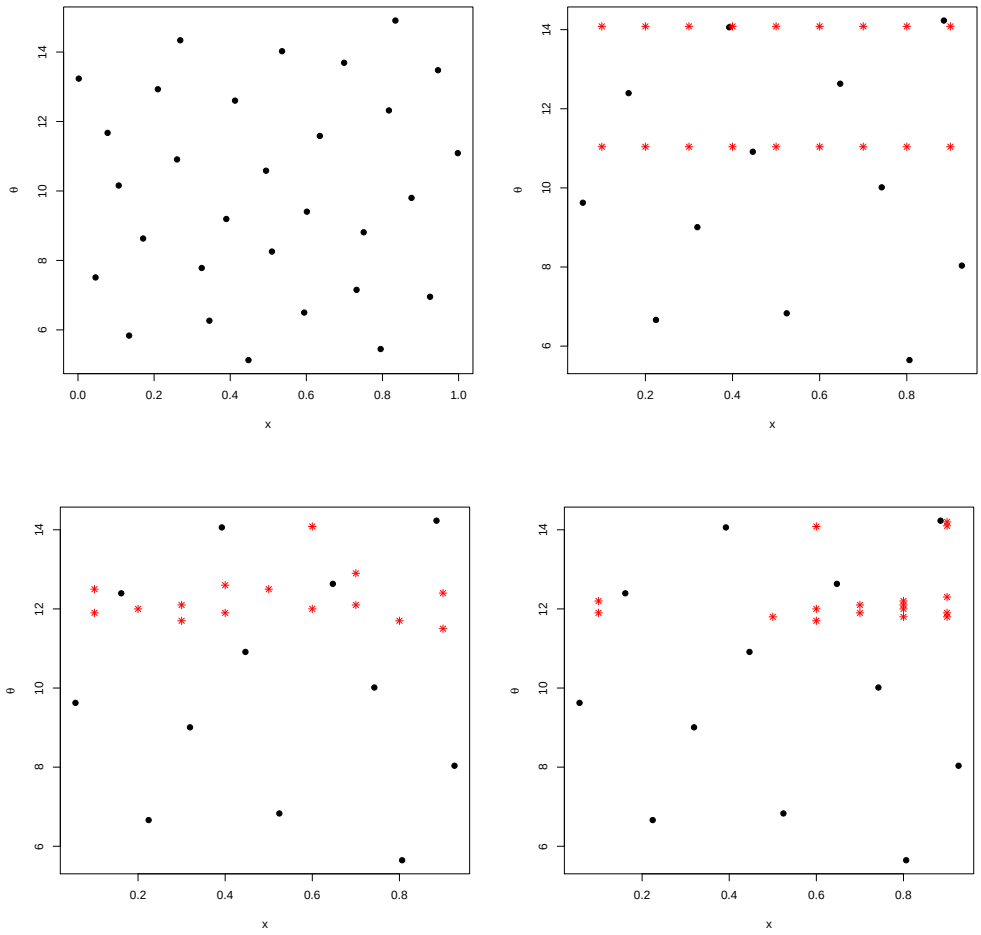
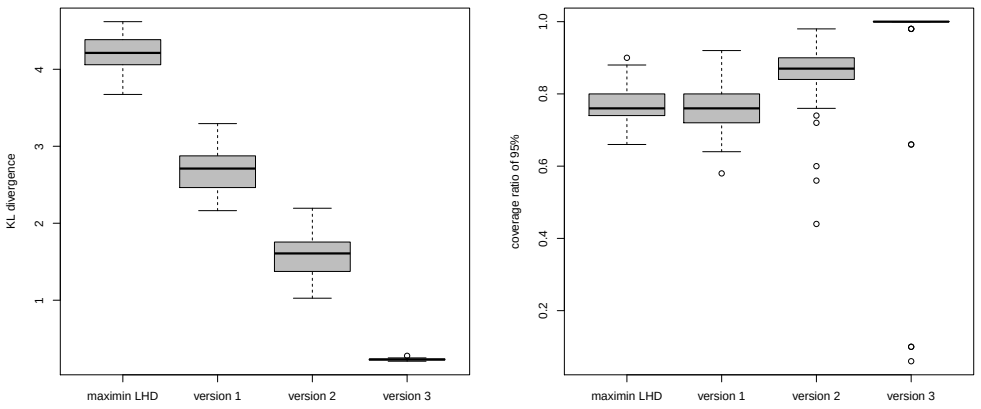


Figure 4.7 – *Case* 2. *Left: boxplots of the KL divergence computed between the target posterior distribution and the surrogate posterior distribution (using the R library FNN). Right: ability of the* 95% *credibility interval to cover the true value* ($\theta = 12$).



This highly non-linear function is used within the field of global sensitivity analysis to assess the efficiency of new methods (Marrel, 2008). For $1 \leq i \leq n = 60$, the field data $\mathbf{z}^f$ are still generated by

$$z_i^f = y_{\boldsymbol{\theta}}(x_i^f) + \epsilon_i\,, \tag{4.36}$$

where $\boldsymbol{\theta} = (0.34, 0.34, 0.34)$ and $\epsilon_i \sim \mathcal{N}(0, 0.05^2)$. Here, both $\mathbf{x}$ and $\boldsymbol{\theta}$ are three-dimensional vectors. In the same spirit as before, we aim at reducing the calibration error between the unknown target posterior (4.6) and the surrogate posterior (4.17). The prior distribution $\pi(\boldsymbol{\theta})$ is chosen as uniform on $[0,1]^3$:

$$\pi(\boldsymbol{\theta}) \propto \mathbf{1}_{[0,1]^3}. \tag{4.37}$$

Let the allocated number of simulations be equal to M = 200. Similar to the 2D example, maximin LHD are compared with the sequential designs. Given the size of field data (n = 60), only Version 2 and Version 3 are performed, both starting from an initial design of size $M_0 = 100$ simulations. As explained in Section 4.3, a discretization G of the parameter space is required for maximizing the EI criterion. Given that the dimension of $\boldsymbol{\theta}$ is larger than in the previous example, the choice of such a grid is more sensitive. Indeed, if G is too coarse, some promising area of the parameter space could be not explored whereas if G is too fine, the computation time will be drastically increased. The solution that we suggest to address this problem consists in maximizing the EI criterion alternatively on several designs, as iterations, so that:

$$G = G_1 \cup G_2 \cup \cdots \cup G_N \tag{4.38}$$

and

$$G_1 \cap G_2 \cap \cdots \cap G_N = \varnothing\,. \tag{4.39}$$

In this example, for simplicity we have chosen N = 2 grids and,

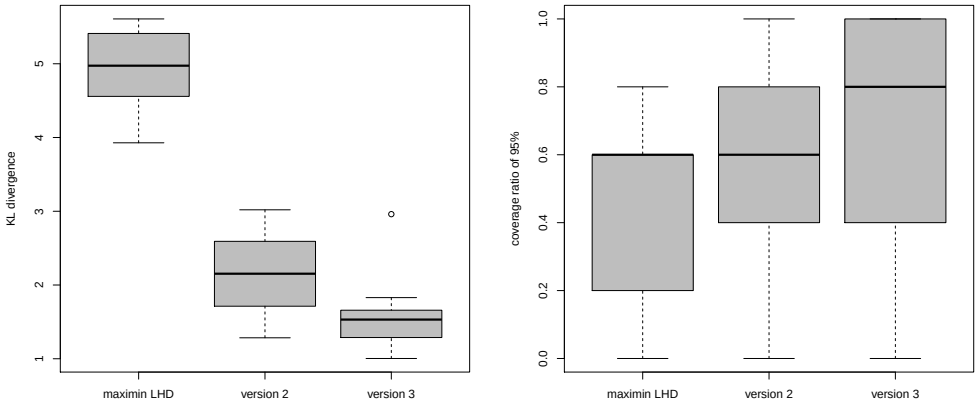
$$G_1 = [0, 0.2, 0.4, 0.6, 0.8, 1] \times [0, 0.2, 0.4, 0.6, 0.8, 1] \times [0, 0.2, 0.4, 0.6, 0.8, 1]$$

and

$$G_2 = [0.1, 0.3, 0.5, 0.7, 0.9] \times [0.1, 0.3, 0.5, 0.7, 0.9] \times [0.1, 0.3, 0.5, 0.7, 0.9]\,.$$

Thus, for odd iterations the EI criterion is maximized over G_1 whereas for even iterations the EI criterion is maximized over G_2. The calibration results are illustrated in Figure 4.8. They look the same as those of the 2D example, which again support the advantage of using a sequential design. Let us see that neither G_1 nor G_2 covers the unknown true value. One can think that a more exhaustive decomposition in (4.38) would make G closer to the true value $\boldsymbol{\theta} = (0.34, 0.34, 0.34)$, perhaps making the results even better. Boxplots for coverage rate have larger variance than in the previous examples because each of the 95% credibility interval of the three marginal posterior distributions needs to cover the true value 0.34.

Figure 4.8 – *Left: boxplots of the KL divergence computed between the target posterior distribution and the surrogate posterior distribution (using the R library FNN). Right: coverage rate. For both figures, boxplots are made over* 50 *calibrations.*



4.5 Conclusion

This paper deals with new adaptive numerical designs for calibrating time-consuming computer codes in a Bayesian setting. For such codes, Bayesian calibration is based on a Gaussian process emulator (GPE) which approximates the code output and thus making the MCMC algorithms practicable. After choosing a design of experiments, the GPE is estimated by using a modular approach which allows us to separate the estimation of GPE parameters from the estimation of the code parameters.

Our contribution consists in taking advantage of the stochastic property of the GPE to build sequentially the design of experiments in such a way that the gap between the posterior distribution based on the GPE (the so-called surrogate posterior distribution) and the posterior distribution based on the code (the so-called target posterior distribution) is the smallest possible in terms of the KL divergence. We have shown it is of a great importance to reduce the uncertainty of the GPE where the density of the target posterior distribution is large. In an objective Bayesian context where no prior expertise is available about the unknown parameters, this goal is equivalent to reduce the uncertainty of the GPE where the sum of squares of the residuals between the code outputs and the field measurements is small. We have thus proposed sequential strategies for building the design of experiments based on the EI criterion designed for the sum of squares of the residuals. Our simulations on toy functions have shown such designs outperform space-filling designs in terms of low value of the KL divergence. However, simulations have been performed in an unbiased framework where there is no discrepancy between the physical system and the computer code. If prior information is available about the shape of the code discrepancy, our algorithms could be applied to a weighted sum of squares function in a similar way.

It may appear surprising that the prior distribution is not taken into account in the EI maximization. That is because the prior distribution is identical both for the surrogate posterior distribution and the target posterior distribution. Furthermore, if we aim at conducting a sensitivity analysis to the choice of the prior distribution, then we would like to be able to perform several calibrations from different prior distributions without having to rebuild a GPE each time.

The main difference between the algorithms presented in this paper and the current use of the EGO algorithm is that the objective function, that is the sum of squares of the residuals, is not modeled by a Gaussian process. Such modeling makes it possible to conduct one-at-a-time sequential strategies where a single simulation $y_{\hat{\boldsymbol{\theta}}_k}(\mathbf{x}^\star)$ is run at each iteration. Two criteria have been suggested to pick up $\mathbf{x}^\star$ among the input field measurements but more sophisticated criteria might be more relevant. In addition, we might be also think about an EGO algorithm which would be designed according to the number of available computers nodes.

Another concern is how to maximize the EI criterion over the parameter space. Because it has no closed form expression, the optimization is performed in a greedy fashion, but free-derivative optimization algorithms could be also used (Conn et al., 2009).

Finally, our EGO algorithms start from a maximin LHD on $\mathscr{X}\times\mathscr{T}$ whereas the posterior distribution only depends on the simulations running in the subspace $\mathbf{X}^f\times\mathscr{T}$. A better strategy might thus be to build an initial design limited to $\mathbf{X}^f\times\mathscr{T}$ where the projection on $\mathscr{T}$ is a maximin LHD.

Proof of Proposition 4.2.0.1

The log-conditional likelihood is referred to as $\ell^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))=\log(\mathscr{L}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}})))$ and the target log-likelihood is referred to as $\ell(\mathbf{z}^f|\boldsymbol{\theta})=\log(\mathscr{L}((\mathbf{z}^f|\boldsymbol{\theta}))$. It is sufficient to prove that

$$|\ell^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))-\ell(\mathbf{z}^f|\boldsymbol{\theta})| \tag{4.40}$$

is uniformly bounded in $\boldsymbol{\theta}$ and the bound tends to zero with $\mathrm{M}\to\infty$. We can decompose (4.40) as

$$\begin{aligned}|\ell^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))-\ell(\mathbf{z}^f|\boldsymbol{\theta})|\leq{}&|\ell^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))-\tilde{\ell}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))|+\\&|\tilde{\ell}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))-\ell(\mathbf{z}^f|\boldsymbol{\theta})|\end{aligned} \tag{4.41}$$

where

$$\tilde{\ell}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))=-\frac{1}{2\lambda^2}(\mathbf{z}^f-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))^{\mathrm{T}}(\mathbf{z}^f-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))-\frac{n}{2}\log 2\pi-n\log\lambda$$

corresponds to the log-conditional likelihood where the function y is replaced with $\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}$ and the covariance matrix of the GPE is neglected.

The second term is bounded as:

$$\begin{aligned}|\tilde{\ell}^{\mathrm{C}}(\mathbf{z}^f|\boldsymbol{\theta},y(\mathbf{D}_{\mathrm{M}}))-\ell(\mathbf{z}^f|\boldsymbol{\theta})| &= \Big|-\frac{1}{2\lambda^2}\Big(||\mathbf{z}^f-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}})||^2-||\mathbf{z}^f-y_{\boldsymbol{\theta}}(\mathbf{X}^f)||^2\Big)\Big|\\&= \Big|-\frac{1}{2\lambda^2}\Big(\sum_{i=1}^{n}(y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}_i^f,\boldsymbol{\theta}))\big(2z_i^f-y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}_i^f,\boldsymbol{\theta})\big)\Big)\Big|.\end{aligned}$$

Let us suppose that the minimax distance, say $(h_{\mathbf{D}_{\mathrm{M}}})_{\mathrm{M}}$, of the designs sequence $(\mathbf{D}_{\mathrm{M}})_{\mathrm{M}}$ tends to 0, namely

$$h_{\mathbf{D}_{\mathrm{M}}}=\max_{(\mathbf{x}',\boldsymbol{\tau}')\in\mathscr{X}\times\mathscr{T}}\min_{(\mathbf{x}_i^{\mathrm{D}},\boldsymbol{\tau}_i^{\mathrm{D}})\in\mathbf{D}_{\mathrm{M}}}\|(\mathbf{x}',\boldsymbol{\tau}')-(\mathbf{x}_i,\boldsymbol{\tau}_i)\|\underset{\mathrm{M}\to\infty}{\longrightarrow}0 \tag{4.42}$$

Then, the uniform bound is deduced from the point-wise bound given for standard radial basis correlation function $\mathrm{C}_{\boldsymbol{\psi}}$ (Schaback, 1995). We can obtain

$$|y_{\boldsymbol{\theta}}(\mathbf{x}_i^f)-\mu^{\mathrm{M}}_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{x}_i^f,\boldsymbol{\theta})|\leq\|y\|_{\mathrm{C}_{\boldsymbol{\psi}}}\cdot\mathrm{G}_{\mathrm{C}_{\boldsymbol{\psi}}}(h_{\mathbf{D}_{\mathrm{M}}}), \tag{4.43}$$

where $\|y\|_{C_\psi}$ is the norm of y in the RKHS associated to C_ψ and $G_{C_\psi}(.)$ tends to 0 when $h_{\mathbf{D}_M}$ tends to 0.

Using the triangle inequality, an upper bound for the first term in (4.41) is written as,

$$|\ell^C(\mathbf{z}^f|\boldsymbol{\theta}, y(\mathbf{D}_M)) - \tilde{\ell}^C(\mathbf{z}^f|\boldsymbol{\theta}, y(\mathbf{D}_M))| \leq \left|\frac{1}{2}\log(|V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n|) - \frac{n}{2}\log(\lambda^2)\right| + \left|\frac{1}{2\lambda^2}\|\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}})\|^2 - \frac{1}{2}(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))^T(V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n)^{-1}(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))\right| \quad (4.44)$$

Then,

$$\begin{aligned}\left|\frac{1}{2}\log(|V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n|) - \frac{n}{2}\log(\lambda^2)\right| &= \left|\frac{1}{2}\log\left(\frac{|V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n|}{(\lambda^2)^n}\right)\right| \\ &\leq \left|\frac{n}{2}\log\left(\frac{\sum_{i=1}^n \left(V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii}/n\right) + \lambda^2}{\lambda^2}\right)\right| \\ &= \left|\frac{n}{2}\log\left(\frac{\sum_i V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii}}{\lambda^2 n} + 1\right)\right|\end{aligned} \quad (4.45)$$

where the inequality of arithmetic and geometric means is used for bounding the determinant of the matrix $V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n$ by a function of the trace. Using again a result in Schaback (1995), we have $V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii} \leq G_{C_\psi}(h_{\mathbf{D}_M})$. It follows that,

$$\left|\frac{n}{2}\log\left(\frac{\sum_i V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii}}{\lambda^2 n} + 1\right)\right| \leq C_y\frac{n}{\lambda^2}G_{C_\psi}(h_{\mathbf{D}_M})$$

where C_y is a constant. Since $(h_{\mathbf{D}_M})_M$ tends to 0 with $M \to \infty$, (4.45) tends to 0 with $M \to \infty$.

For the second term (4.44), we use the series expansion $(A + \mathbf{I}_n)^{-1} = \mathbf{I}_n + \sum_{i=1}^\infty (-1)^i A^i$, valid when $||A|| < 1$. It can thus be applied to $V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})\lambda^{-2}$ because both inequalities $V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii} \leq G_{C_\psi}(h_{\mathbf{D}_M})$ and $|V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ij}| \leq \sqrt{V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{ii}V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})_{jj}}$ ensure its norm is lower than 1 once M is large enough. Therefore,

$$\left|\frac{1}{2\lambda^2}\|\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}})\|^2 - \frac{1}{2}(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))^T(V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta}) + \lambda^2\mathbf{I}_n)^{-1}(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))\right| = \left|\frac{1}{2\lambda^2}(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))^T\left(\sum_{i=1}^\infty (-1)^i[V^M_{\boldsymbol{\Psi},\sigma^2}(\boldsymbol{\theta})\lambda^{-2}]^i\right)(\mathbf{z}^f - \mu^M_{\boldsymbol{\beta},\boldsymbol{\Psi}}(\mathbf{D}_{\boldsymbol{\theta}}))\right|$$

which tends to 0 with $M \to \infty$, that completes the proof of the uniform convergence of (4.40) to 0 with $M \to \infty$.

Bibliography

Bachoc, F. (2014). Asymptotic analysis of the role of spatial sampling for covariance parameter estimation of Gaussian processes. *Journal of Multivariate Analysis*, 125:1–35. 81

Bastos, L. and O'Hagan, A. (2008). Diagnostics for Gaussian process emulators. *Technometrics*, 51(4):425–438. 81

Bayarri, M., Berger, J., Paulo, R., Sacks, J., Cafeo, J., Cavendish, J., Lin, C., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49(2):138–154. 77, 82

Box, G. E. P. and Tiao, G. T. (1973). *Bayesian Inference in Statistical Analysis*. Addison-Wesley, Reading. 85

Brynjarsdottir, J. and O'Hagan, A. (2014). Learning about physical parameters: The importance of model discrepancy. *Inverse Problems*, 30(11):3251–3269. 77

Campbell, K. (2006). Statistical calibration of computer simulations. *Reliability Engineering and System Safety*, 91(10-11):1358–1363. 77

Conn, A., Sheinberg, K., and Vicente, L. (2009). *Introduction to Derivative-Free Optimization*. MOS SIAM Series on Opimization, Reading. 95

Cover, T. and Thomas, J. (1991). *Elements of Information Theory*. Wiley-Interscience. 83

Cox, D., Park, J., and Clifford, E. (2001). A statistical method for tuning a computer code to a data base. *Computational Statistics and Data Analysis*, 37(1):77–92. 77, 81

Craig, P., Goldstein, M., Rougier, J., and Seheult, A. (2001). Bayesian forecasting for complex systems using computer simulators. *Journal of the American Statistical Association*, 96(454):717–729. 83

Craig, P., Goldstein, M., Seheult, A., and Smith, J. (1997). *Pressure Matching for Hydrocarbon Reservoir History: A Case Study in the Use of Bayes Linear Strategies for Large Computer Experiments*, volume 121 of *Lecture Notes In Statistics*. Springer-Verlag. 77

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the Statistical Association*, 86(416):953–963. 80

Ellis, N. and Maitra, R. (2007). Multivariate Gaussian simulation outside arbitrary ellipsoids. *Journal of Computational and Graphical Statistics*, 16(3):692–708. 87

Fang, K., Li, R., and Sudijianto, A. (2006). *Design and modeling for computer experiments*. Computer Science and Data Analysis. Chapman & Hall. 81

Gang, H., Santner, T., and Rawlinson, J. (2009). Simultaneous determination of tuning and calibration parameters for computer experiments. *Technometrics*, 51(4):464–474. 78

Ginsbourger, D. (2009). *Multiples Métamodèles pour l'approximation et l'optimisation de fonctions numériques multivariables*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne. 84

Gramacy, R., Bingham, D., Holloway, J., Grosskopf, M., Kuranz, C., Rutter, E., Trantham, M., and Drake, P. (2014). Calibrating a large computer experiment simulating radiative shock hydrodynamics, forthcoming. 77, 81

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J., and Ryne, R. (2004). Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466. 77, 79, 81, 82

Janusevskis, J. and Le Riche, R. (2013). Simultaneous kriging-based estimation and optimization of mean response. *Journal of Global Optimization*, 55(2):313–336. 78

Jones, D., Schonlau, M., and Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492. 78, 84, 86

Joseph, V. and Melkote, S. (2009). Statistical adjustments to engineering models. *Journal of Quality Technology*, 41(4):362–375. 79, 85

Kass, R. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91(435):1343–1370. 85

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society, Series B, Methodological*, 63(3):425–464. 77, 78, 79, 82

Koehler, J. and Owen, A. (1996). Computer experiments. *In Design and Analysis, Handbook of Statistics, Vol.13*, pages 261–308. 79

Kumar, A. (2008). *Sequential calibration of computer models*. PhD thesis, The Ohio State University. 78, 79

Liu, F., Bayarri, M., and Berger, J. (2009). Modularization in bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150. 81, 82

Loeppky, D., Bingham, D., and Welch, W. (2006). Computer model calibration or tuning in practice. Technical report. 77

Marrel, A. (2008). *Mise en oeuvre et utilisation du méta-modèle processus gaussien pour l'analyse de sensibilité de modèles numériques*. PhD thesis, CEA Cadarache. 93

Pratola, M., Sain, S., Bingham, D., Wiltberger, M., and Rigler, E. (2013). Fast sequential computer model calibration of large nonstationary spatial-temporal processes. *Technometrics*, 55(2):232–242. 78

Pronzato, L. and Muller, W. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701. 83

Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. the MIT press. 81

Robert, C. and Casella, G. (1998). *Monte Carlo Statistical Methods*. Springer-Verlag. 80

Roy, C. and Oberkampf, W. (2011). A comprehensive framework for verification, validation and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2131–2144. 77

Sacks, J., W.J., W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423. 80

Saltelli, A., Chan, K., and Scott, E. (2000). *Sensitivity Analysis*. Wiley, New York. 77, 91

Santner, T., Williams, B., and Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag. 81, 82

Schaback, R. (1995). Error estimates and condition numbers for radial basis function interpolation. *Advances in Computational Mathematics*, 3(3):251–264. 83, 95, 96

Sheil, J. and Muircheartaigh, I. (1977). The distribution of non-negative quadratic forms in normal variables. *Journal of the Royal Statistical Society*, 26(1):92–98. 87

Stein, M. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, New York. 80, 81

Vazquez, E. and Bect, J. (2010). Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095. 84

Wong, R., Storlie, C., and Lee, T. (2014). A frequentist approach to computer model calibration. Technical report, Iowa State University. 77, 85

Chapter 5

Validation of a computer code for the energy consumption of a building, with application to optimal electric bill pricing

Contents

Abstract The paper displays the steps for the validation of a thermal computer code which is used to predict the electric power delivered inside an experimental cell. The validation study will be performed with the aim of the guarantee of energy performance, which will be addressed using Bayesian decision theory.

Based on power field measurements which are collected inside the cell over a time period of 7 days, the code is calibrated using a statistical approach for reducing the epistemic uncertainty affecting some code parameters, including the albedo, the thermal bridge factor and the convective coefficient. In a second stage, the validation of the code is conducted by propagating The uncertainty to the code output in order to predict the average electric power delivered inside the cell over the 7 days. The last part of the work addresses a decision problem whereby an energy supplier has to commit for an overall energy consumption forecast to customers. Based on Bayesian decision theory, some optimal forecasts are obtained according to the kind of energy contract subscribed by customers.

Keywords: Validation, Bayesian calibration, Bayesian decision theory.

Résumé Cet article illustre les étapes successives permettant la validation d'un code de calcul utilisé en thermique du bâtiment pour prédire la puissance électrique injectée à l'intérieur d'une cellule expérimentale, avec en ligne de mire l'objectif de la garantie de performance énergétique que nous abordons sous l'angle de la théorie bayésienne de la décision.

Sur la base des mesures de puissance effectuées au cours d'une période de temps de 7 jours, le calage du code de calcul permet dans un premier temps de réduire l'incertitude épistémique affectant les paramètres du code, dont l'albedo, le coefficient quantifiant les ponts thermiques, ainsi que le facteur de convection du dispositif de régulation d'air à l'intérieur de la cellule. Ensuite, le code est validé en propageant à la réponse du code l'incertitude *a posteriori* affectant ces paramètres, puis en calculant l'incertitude de prédiction affectant la puissance moyenne prédite sur la période des 7 jours. Enfin, en s'appuyant sur cette étude de validation, nous proposons une solution à un problème de statistique décisionnelle qui consiste à calculer la prévision de consommation énergétique moyenne sur laquelle un fournisseur d'électricité devra s'engager auprès de ses clients. En utilisant la théorie de la décision bayésienne, des estimateurs optimaux sont calculés en fonction des différentes fonctions de coût proposées.

Mots-clés: Validation, Calage bayésien, Théorie de la décision bayésienne.

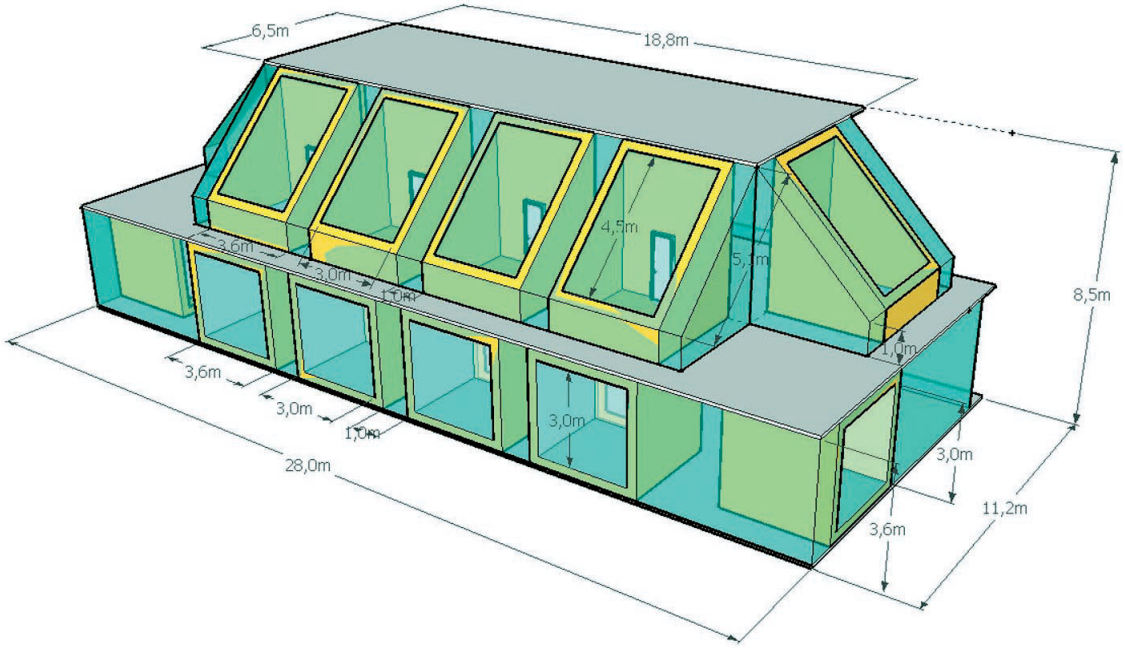
5.1 Introduction

Today, more and more green buildings are constructed to maintain a low energy consumption with the objective of dividing by four the CO2 emissions in 2050. In this context, the guarantee of building performance is now a challenging issue which can be tackled by using computer codes. At the conception stage, they should be able to predict the actual energy performance of such buildings.

In sciences and engineering, complex physical systems are mainly studied by means of computer codes, either because the physical experiments are unfeasible or because they are economically too expensive to be conducted. Unfortunately, large differences are often observed between the physical measurements and the code predictions, putting in question how well the code is capable of reproducing the physical system. That is why, the agreement between these two data sets should be checked to conclude whether or not the code can be used as a surrogate of the reality. This task, called *validation*, has been already addressed in the field of building energy simulation, but only through graphical comparisons. For instance, Bontemps et al. (2013) actually checked the graphical agreement between some temperature measurements and some corresponding code predictions. Although easy to use, their method only remains qualitative and will not provide the causes of a poor fitting. Many works dedicated to the validation therefore propose to assess the accuracy of the code predictions by taking into account all the uncertainties, including both *aleatory* and *epistemic* uncertainty (Roy and Oberkampf, 2011). The aleatory uncertainty is essentially due to both weather conditions and inhabitants profile. The epistemic uncertainty is strongly different by nature from the aleatory uncertainty. It stems from a lack of knowledge about the value of some code parameters and possibly about the source code itself. Spitz (2012) treats the aleatory uncertainty in the conception stage by conducting a local sensitivity analysis, then a global sensitivity analysis in order to identify what are the factors which have the most impact on the code predictions. In this paper, we rather focus on the treatment of the epistemic uncertainty.

We have at our disposal a thermal computer code which is constructed with the Dymola software, itself deriving from the Modelica language. This code is dynamic and can thus compute some predictions of the electric power delivered inside an experimental cell over a time period as a function of input variables consisting of the size of the cell, the inside temperature and the weather conditions (such as the outside temperature, the wind speed and so forth). We assume no aleatory uncertainty taints these inputs because they have been measured precisely on and around the cell over time. The decision of performing here an *uncertainty analysis* of this code is motivated by the uncertainty tainting the value of some of its parameters, including the thermal bridge factor, the albedo and the convective factor associated with the air control system inside the cell (see Figure 5.2). The quantification of such a parametric uncertainty is called *calibration* and will be the first stage of our study (Campbell, 2006). The parametric uncertainty is epistemic because it reflects a lack of knowledge about the value of the parameters. It is encoded in this work as a probability distribution which combines a *prior* belief about the value of the uncertain parameters from a Bayesian point of view with the information provided by the data (Bernardo and Smith, 1994). Let us stress the epistemic uncertainty can also arise from a lack of knowledge either about the degree of adequacy between the code and the physical phenomenon (Kennedy and O'Hagan, 2001), or affecting the parameters of a probability distribution encoding the aleatory uncertainty of some input variables (Pasanisi et al., 2012). The second stage of our study will consist in propagating the parametric uncertainty to the code output, then to the average electric power delivered inside

Figure 5.1 – EDF BESTLab platform.



the cell over the time period. This stage is closely related with the validation of the code. In the last stage, we deal with a decision problem whereby an energy supplier has to commit for an overall consumption forecast to customers with the help of Bayesian decision theory (Bernardo and Smith, 1994; Ulmo and Bernier, 1973). Some Bayes estimators are calculated thanks to both the probability distribution of the average electric power calculated previously and an utility function designed to assess the economical consequences of each estimate based on the cost of the electricity production.

In Section 5.2, the main tasks of our study, namely calibration, validation and decision theory are detailed towards the field of energy buildings. In Section 5.3, after assuming a statistical model between the code outputs and the power field measurements, the calibration of the thermal code is performed using a Bayesian approach. In Section 5.4, the validation of the code is conducted by calculating the uncertainty affecting the average electric power delivered inside the cell over the time period. In Section 5.5, several optimal consumption forecasts are calculated using Bayesian decision theory. The conclusions of this work are given in Section 5.6.

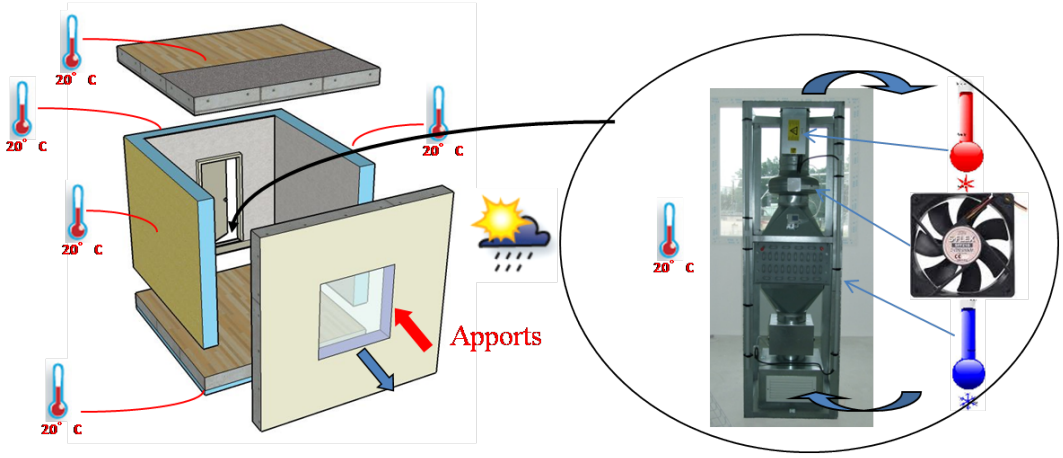
5.2 Overview of the study

The code based on the Dymola software predicts the electric power $y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)$ delivered inside an experimental cell as a function of a vector $\mathbf{x}_t$ of physical inputs at the time step t, including the cell characteristics, the temperature inside the cell and weather conditions such as wind speed, the outside temperature and so forth. In addition, this code depends on a vector of fixed parameters $\boldsymbol{\theta}$ and can be thus denoted by the function

$$y : (\mathbf{x}_t, \boldsymbol{\theta}) \in \mathscr{X} \times \mathscr{T} \subset \mathbb{R}^n \times \mathbb{R}^p \longrightarrow y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t) \in \mathbb{R}. \tag{5.1}$$

It is also possible to tune the reverse code to have the temperature as a function of the electric power as done in a past study (Bontemps et al., 2013). The dimension of $\boldsymbol{\theta}$ is $p =$ 193 including both some physical parameters such as the albedo, the convective factor,

Figure 5.2 – Left: the experimental cell. Right: the air conditioning system inside.



and many design variables such as the floor surface, the window width and so forth. The experimental cell is located in the EDF site *Les Renardières* at about 75 km southeast of Paris. It is one of the twelve cells making up the experimental BESTLab platform. Only the wall equipped with a window is in contact with the outside (see Figure 5.2). All complete features of the cell are detailed in Bontemps et al. (2013).

The power field measurements are denoted by the vector $z(\mathbf{x}_t)(t)$. They are collected inside the cell every 5 minutes over the time period of 7 days. Let $\mathrm{P}(\mathbf{x}_t)(t)$ be the actual electric power inside the cell at time t. Owing to the uncertainty inherent in the experimental protocol, the power measurements $z(\mathbf{x_t})(\mathrm{t})$ are not exactly equal to $\mathrm{P}(\mathbf{x}_t)(t)$. The link between both can be established through the statistical equation

$$z(\mathbf{x}_t)(t) = \mathrm{P}(\mathbf{x}_t)(t) + \epsilon_t \tag{5.2}$$

where

$$\epsilon_t \underset{i.i.d.}{\sim} \mathcal{N}(0, \lambda^2)$$

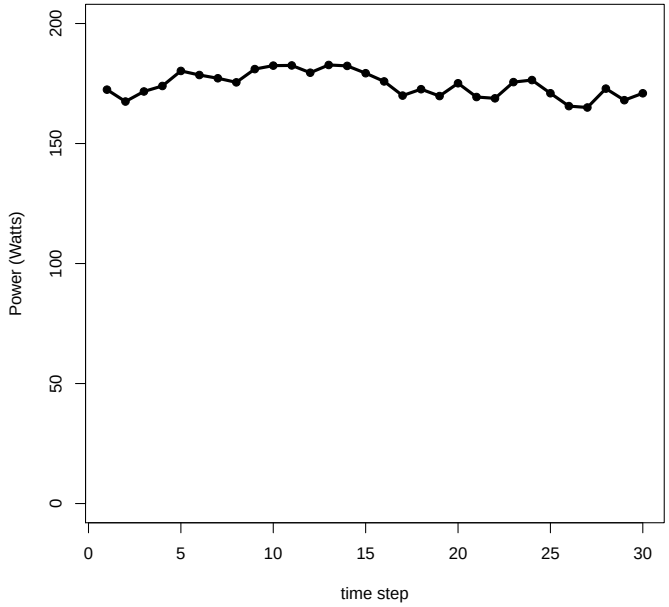
is a zero mean Gaussian random variable which includes the residual variability (Kennedy and O'Hagan, 2001) as well as the measurement error of all physical quantities. In this paper, computations are based on a vector of mean power data $\mathbf{z}^f$ instead of the vector of complete power measurements, about four per day instead of one per 5 minutes, yielding 30 data over the 7 days (see Figure 5.3):

$$\mathbf{z}^f := (z(\mathbf{x}_1)(1), \cdots, z(\mathbf{x}_{30})(30)). \tag{5.3}$$

Calibration Code calibration consists in reducing the epistemic uncertainty tainting the code predictions. The thermal code depends on the vector $\boldsymbol{\theta}$ of physical parameters which is typically set to an unchanged value before running the code. However, the vector $\boldsymbol{\theta}$ may be uncertain, either because it is non-measurable in the field or because it has no counterpart in the physical system (Loeppky et al., 2006).

In calibration, it is important to keep in mind that a computer code might be an imperfect representation of the physical system. The thermal code should be thus considered as a more or less accurate mathematical approximation of the thermal behaviour inside the

Figure 5.3 – *The* 30 *averaged power measurements. The time step is approximately* 5 *hours and* 30 *minutes.*



experimental cell. This second source of epistemic uncertainty is called *code uncertainty* (Kennedy and O'Hagan, 2001).

In this paper, the calibration of the thermal code is conducted following a Bayesian statistical approach which needs two ingredients:

— the *prior* density $\pi(\boldsymbol{\theta})$ encoding the uncertainty as a *prior* belief in favor of some values of $\boldsymbol{\theta}$, which are more probable than others. If no information is known about that, a flat *prior* distribution should be taken.

— a statistical model which links the available field measurements $\mathbf{z}^f$ with the code outputs. This equation gives a likelihood function $\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta},\mathbf{p})$ where $\mathbf{p}$ is a vector of nuisance parameters attached to the model such as those specifying the error structure between the code outputs and the field measurements. A testing procedure shall be used in order to avoid falsification of the chosen model according to its intended use.

In a Bayesian framework, the uncertainty affecting both the vector of code parameters $\boldsymbol{\theta}$ and the vector of parameters $\mathbf{p}$ is updated thanks to the Bayes formula, given by:

$$\begin{aligned}\pi(\boldsymbol{\theta},\mathbf{p}|\mathbf{z}^f) &= \frac{\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta},\mathbf{p})\pi(\boldsymbol{\theta},\mathbf{p})}{\int_{\boldsymbol{\theta},\mathbf{p}}\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta},\mathbf{p})\pi(\boldsymbol{\theta},\mathbf{p})\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{p}} \\ &\propto \mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta},\mathbf{p})\pi(\boldsymbol{\theta},\mathbf{p}) \end{aligned} \tag{5.4}$$

Given that the dimension of $\boldsymbol{\theta}$ is very large ($p = 193$), the calibration of the thermal code is not directly feasible. Previous to calibration, the parameters having the most influence on the power predictions should be identified thanks to a sensitivity analysis (Saltelli et al., 2000). Bontemps et al. (2013) performed a local sensitivity analysis of $\boldsymbol{\theta}$ so that its dimension was downsized from $p = 193$ to $p = 13$, then a global sensitivity analysis of $\boldsymbol{\theta}$ so that its dimension was again downsized from $p = 13$ to $p = 3$. The local sensitivity indices have been obtained versus time via the one at a time method which consists in calculating the

partial derivative of the code around a nominal value of the parameters. Then, the global sensitivity indices have been derived from a polynomial chaos expansion of the code output. Finally, the three scalar parameters which yield the greatest impact on the electric power are:

— $\theta_1 \in [0,1]$ which is the albedo factor,
— $\theta_2 > 0$ which encodes the thermal bridges,
— $\theta_3 > 0$ which is the convective factor of the air conditioning system.

Validation This task is part of the V&V (Verification and Validation) framework aiming at quantifying the accuracy of the code predictions (AIAA, 1998). Verification consists of checking if all bugs inside the source code have been removed and assessing the discretization error when the code is constructed as a finite element solution (Roache, 1998). In this paper, we are not concerned with Verification although this task should be addressed before any validation study.

Validation has to ensure that the mathematical representation that underlies the code is an acceptable representation of the physical system. Over years, several frameworks for this task have been applied to perform much better than the basic graphical comparisons between code outputs and field measurements, thanks to a proper quantification of the uncertainty affecting them. For instance, Roy and Oberkampf (2011) constructed some measures of validation based on statistical tests by taking into account that the computer code may be exposed to different errors by nature, including the intrinsic variability of its inputs (aleatory uncertainty) and the parametric uncertainty (epistemic uncertainty). Wang et al. (2009) proposed a Bayesian validation framework where the discrepancy between the code and the field measures is modeled by a random Gaussian process. In our study, the aleatory uncertainty is negligible because $\mathbf{x}_t$ is measured precisely at each time step t over the time period and the measurement error attached to it is included in the noise ϵ_t (5.2). On the other hand, the power predictions are affected by both the value of the code parameters and the adequacy of the code itself to the thermal system. The validation stage is thus closely related with the calibration stage. Bayarri et al. (2007) tackled these tasks together. In their framework, the epistemic uncertainty is first quantified including the calibration stage, then it is is propagated to the code output. In the next sections, the validation of the thermal code is performed in a similar way.

Lastly, a validation study should be carried out keeping in mind the intended use of the code. The code predictions should be actually judged sufficiently accurate according to the estimation of a so-called quantity of interest such as the standard deviation of the injected power inside the cell or the probability that the injected power exceeds an upper bound over the time period. In this study, we aim at assessing the uncertainty affecting the average power $\phi(\mathrm{P}(\mathbf{x}_1)(1),\cdots,\mathrm{P}(\mathbf{x}_{30})(30))$ delivered inside the cell over the time period. Hence,

$$\phi : (\mathrm{P}(\mathbf{x}_1)(1),\cdots,\mathrm{P}(\mathbf{x}_{30})(30)) \longrightarrow \bar{\mathrm{P}} = \frac{1}{30}\sum_{t=1}^{30} \mathrm{P}(\mathbf{x}_t)(t). \tag{5.5}$$

It follows that the uncertainty affecting $\bar{\mathrm{P}}$ derives from the uncertainty affecting $\mathrm{P}(\mathbf{x}_t)(t)$ at each time step over the period.

Decision theory The link between *plug-in* estimation to decision under uncertainty within industrial studies has been carefully studied (Pasanisi et al., 2012). We recall in this

paragraph the ingredients of this approach applied to the field of energy buildings. The estimation of an uncertain quantity in a Bayesian context provides a measure of the uncertainty under the form of a probability distribution. If a point estimate is required, a Bayes estimator should be calculated based on both the probability distribution of the quantity and a *cost function* which assesses the economical consequences of choosing a point estimate rather than the unknown true value of the quantity. Let us assume a probability distribution $\pi(\bar{\mathrm{P}})$ of the average consumption is available (see Section 4). Plug-in estimation could be done by calculating the *posterior* maximum, namely the mode of $\pi(\bar{\mathrm{P}})$ but it is not context dependent. In view of anticipating economical consequences, such a point estimate may not be the best choice to commit for an energy consumption forecast to customers.

Let d be the quantity to be estimated. The cost function is denoted by $\mathrm{C}(d,\bar{\mathrm{P}})$ and should measure the economical consequences induced by the choice of d instead of the true average consumption $\bar{\mathrm{P}}$ (of course unknown). Then, a Bayes estimate $\hat{d}$ minimizes the average cost against the probability distribution of $\bar{\mathrm{P}}$, that is

$$\hat{d} = \underset{d}{\operatorname{argmin}} \int_{\bar{\mathrm{P}}} \mathrm{C}(d,\bar{\mathrm{P}})\pi(\bar{\mathrm{P}})\mathrm{d}\bar{\mathrm{P}}. \tag{5.6}$$

A cost function is sometimes replaced by a *utility function* function as an assessment of the profit instead of the loss. In this case, a Bayes estimate $\hat{d}$ maximizes the average utility, that is

$$\hat{d} = \underset{d}{\operatorname{argmax}} \int_{\bar{\mathrm{P}}} \mathrm{U}(d,\bar{\mathrm{P}})\pi(\bar{\mathrm{P}})\mathrm{d}\bar{\mathrm{P}}. \tag{5.7}$$

In Section 5, two utility functions $\mathrm{U}(d,\mathrm{P})$ are presented according to the kind of energy contract proposed by the energy supplier.

5.3 Calibration of the dynamic thermal code

Code Calibration According to Section 1, calibration of the thermal code requires to assume a statistical model between the code output and the power field measurements. Let us assume that the model discrepancy is negligible between the code outputs and the field measurements. Hence,

$$\exists \boldsymbol{\theta} \in \mathscr{T} \ ; \ y_{\boldsymbol{\theta}}(\mathbf{x}_t)(\mathrm{t}) = \mathrm{P}(\mathbf{x}_t)(t). \tag{5.8}$$

Then, Equation (5.2) implies:

$$z(\mathbf{x}_t)(t) = y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t) + \epsilon_t \tag{5.9}$$

where

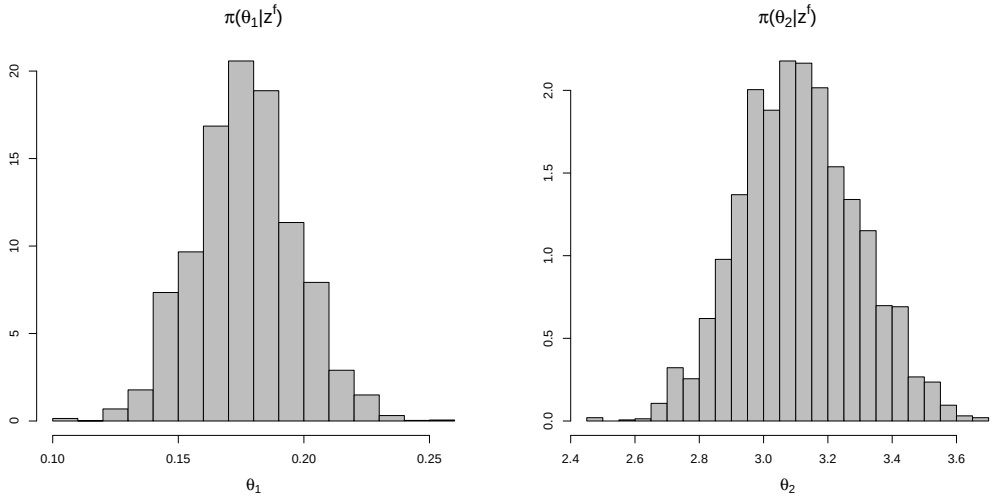
$$\epsilon_t \underset{i.i.d.}{\sim} \mathscr{N}(0,\lambda^2).$$

As the variance λ^2 is unknown, it is estimated together with $\boldsymbol{\theta}$. Since the noise ϵ_t is Gaussian, the likelihood [1] is written as

$$\mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta},\lambda^2) = \frac{1}{(\sqrt{2\pi}\lambda)^n} \exp\Big[-\frac{1}{2\lambda^2}\mathrm{SS}(\boldsymbol{\theta})\Big] \tag{5.10}$$

1. the distribution of $\mathbf{z}^f$ conditional to $(\boldsymbol{\theta},\lambda^2)$

Figure 5.4 – *Left : posterior distribution of* θ_1. *Right : posterior distribution of* θ_2.



where

$$SS(\boldsymbol{\theta}) = ||\{z(\mathbf{x}_t)(t) - y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)\}_t||^2 \tag{5.11}$$

is the sum of squares of the residuals between the power field measurements and the code outputs. From the Bayes formula (5.4) where $p := \lambda^2$,

$$\pi(\boldsymbol{\theta}, \lambda^2|\mathbf{z}^f) \propto \mathscr{L}(\mathbf{z}^f|\boldsymbol{\theta}, \lambda^2)\pi(\boldsymbol{\theta}, \lambda^2) \tag{5.12}$$

As $y_{\boldsymbol{\theta}}(\mathbf{x})$ is non-linear, the *posterior* distribution (5.12) has no analytic form. It is sampled using a Metropolis-Hastings algorithm (Robert and Casella, 1998) where the *prior* distribution $\pi(\boldsymbol{\theta})$ is chosen as a product of independent uniform distributions and the Jeffrey non-informative *prior* is specified on the variance λ^2:

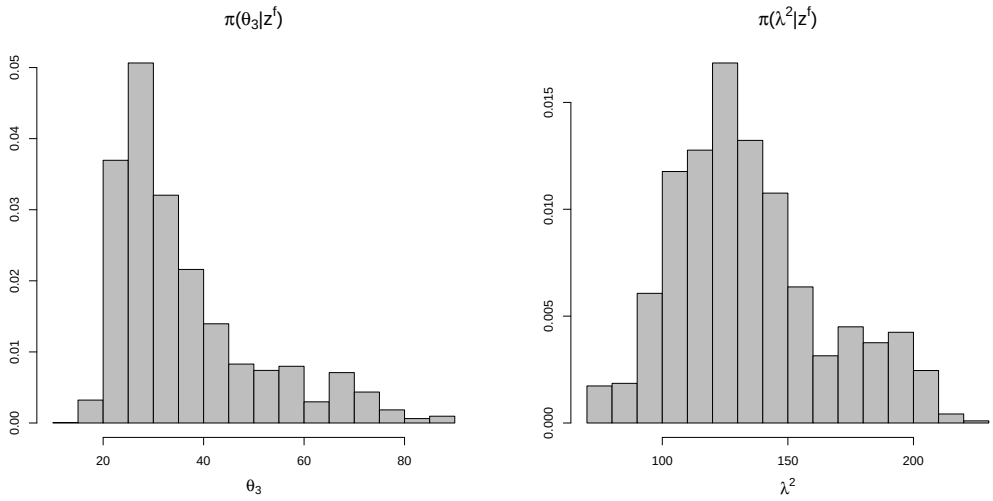
$$\pi(\boldsymbol{\theta}, \lambda^2) = \pi(\theta_1)\pi(\theta_2)\pi(\theta_3)\pi(\lambda^2) \tag{5.13}$$

where

$$\pi(\theta_1) = \frac{\mathbf{1}_{[0,1]}(\theta_1)}{0.1} \quad \pi(\theta_2) = \frac{\mathbf{1}_{[0,100]}(\theta_2)}{100} \quad \pi(\theta_3) = \frac{\mathbf{1}_{[0,100]}(\theta_3)}{100}, \quad \pi(\lambda^2) \propto \frac{1}{\lambda^2}$$

Figures 5.4 and 5.5 show the densities of the marginal *posterior* distributions $\pi(\boldsymbol{\theta_i}|\mathbf{z}^f)$ and $\pi(\lambda^2|\mathbf{z}^f)$. In the next section, a validation framework of the code is presented. It is based on the fact that the electric power P_t at time step t is assumed equal to the code output $y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)$ (see Equation (5.8)) and thus depends on $\boldsymbol{\theta}$.

Figure 5.5 – *Left : posterior distribution of* θ_3. *Right : posterior distribution of* λ^2.



5.4 Validation of the dynamic thermal code

Code validation As detailed in Section 5.2, validation of the thermal code consists in assessing the uncertainty affecting the average power $\bar{P}$ which is delivered inside the cell over the time period. This process requires the followings steps:

— **step 1**: sample $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M$ from the posterior distribution $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$. This step was achieved in Section 5.3,

— **step 2**: run the code over the M samples $\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M$. From here, a probability distribution $\pi(y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t))$ for the electric power at each time step t can be calculated by means of $y_{\boldsymbol{\theta}_1}(\mathbf{x}_t)(t), \cdots, y_{\boldsymbol{\theta}_M}(\mathbf{x}_t)(t)$. A much-used point estimate is the predictive estimator, given by

$$\int y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)\pi(\boldsymbol{\theta}|\mathbf{z}^f)\mathrm{d}\boldsymbol{\theta} \tag{5.14}$$

which is illustrated by the blue line in Figure 5.6.

— **step 3**: compute $\pi(\bar{P})$ thanks to some samples $\bar{y}_{\boldsymbol{\theta}_i}$

$$\bar{y}_{\boldsymbol{\theta}_i} = \frac{1}{30}\sum_{t=1}^{30} y_{\boldsymbol{\theta}_i}(\mathbf{x}_t)(t)$$

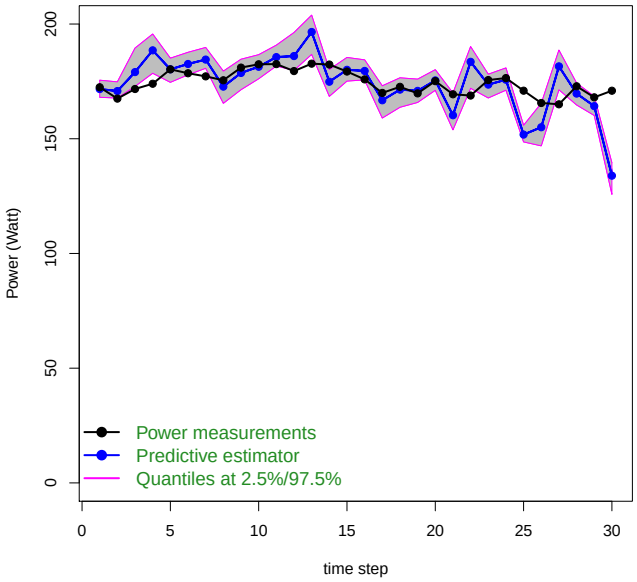
where the upper bar denotes the mean over the time period. Then, practitioners should now decide whether or not the uncertainty on $\bar{P}$ is not too large in view of its intended use, for instance by calculating the quantile at 2.5% and at 97.5% of $\bar{P}$. From $\pi(\bar{P})$ (see Figure 5.7), we obtain the values 170.63 Watts and 177.95 Watts respectively.

To summarize, this procedure consists of a calibration stage (step 1), then two consecutive stages of uncertainty propagation (step 2 and 3). Therefore, the results of both steps 2 and 3 only depend on the probability distribution $\pi(\boldsymbol{\theta}|\mathbf{z}^f)$, which depends itself on the statistical model chosen for calibration. In the literature dedicated to code calibration, another statistical equation is often set out under the form

$$y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t) = \mathrm{P}(\mathbf{x}_t)(t) + b(\mathbf{x}_t)(t) \tag{5.15}$$

This is equivalent to assume there exists no value of $\boldsymbol{\theta}$ making a perfect agreement between $\mathrm{P}(\mathbf{x}_t)(t)$ and $y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)$ (Kennedy and O'Hagan, 2001). The function b is usually called code discrepancy. Joint estimation of $\boldsymbol{\theta}$ and $b(\mathbf{x})$ has been performed in the work

Figure 5.6 – *Probability distribution* $\pi(\mathrm{P}_t)$ *of the electric power delivered inside the cell at each time step of the period.*



of Bayarri et al. (2007) despite confounding between both of them. In what follows, we check there is no evidence for b in order to justify the previous calibration work. A statistical testing is hereafter proposed based on the *posterior* predictive p-value (Meng, 1994).

Statistical testing To check whether or not the statistical equation (5.9) is realistic, a testing procedure is conducted hereafter, based on the computation of a *posterior* predictive p-value. This Bayesian testing procedure can take into account the distribution of a vector of nuisance parameters such as here $\boldsymbol{\theta}$ and λ^2 (Meng, 1994). In this paper, this tool will be used to make a posterior assessment of the goodness of fit of the statistical model (5.9) to the power date $\mathbf{z}^f$ (Gelman et al., 1996). Let us consider the $\mathscr{X}^2$ discrepancy

$$\mathscr{X}^2(\mathbf{z}^f, \boldsymbol{\theta}, \lambda^2) = \sum_{t=1}^{30} \frac{\left(z(\mathbf{x}_t)(t) - y_{\boldsymbol{\theta}}(\mathbf{x}_t)(t)\right)^2}{\lambda^2}. \tag{5.16}$$

Then, the *posterior* predictive p-value based on this discrepancy is denoted by p_{B}. It is written as

$$p_{\mathrm{B}} = \mathbb{P}[\mathscr{X}^2(\mathbf{z}^f_{rep}, \boldsymbol{\theta}, \lambda^2) > \mathscr{X}^2(\mathbf{z}^f, \boldsymbol{\theta}, \lambda^2) | \mathbf{z}^f] \tag{5.17}$$

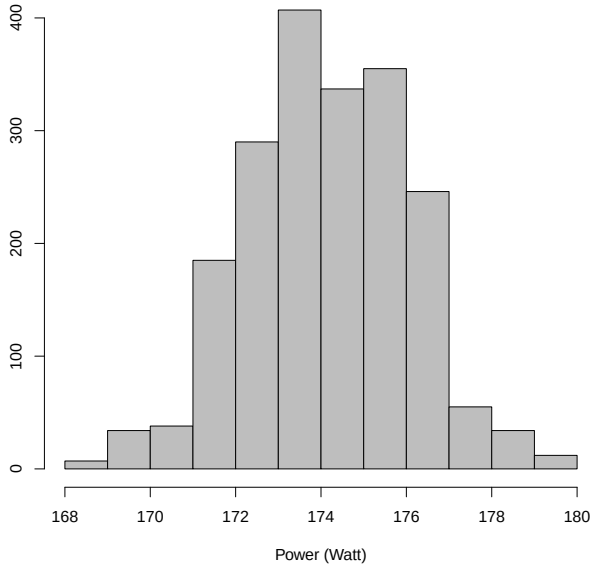
where $\mathbf{z}^f_{rep}$ is a vector of replicated data as the future data that could be produced from the model (5.9). The probability in (5.17) is taken over the joint *posterior* distribution of $(\mathbf{z}^f_{rep}, (\boldsymbol{\theta}, \lambda^2))$ of density,

$$f(\mathbf{z}^f_{rep}, (\boldsymbol{\theta}, \lambda^2)) = f(\mathbf{z}^f_{rep} | (\boldsymbol{\theta}, \lambda^2)) \pi((\boldsymbol{\theta}, \lambda^2) | \mathbf{z}^f) \tag{5.18}$$

Hence, (5.17) can be estimated via Monte Carlo simulation by using samples $(\mathbf{z}^f_{rep,i}, \boldsymbol{\theta}_i, \lambda^2_i)$. First, $(\boldsymbol{\theta}_i, \lambda^2_i)$ is simulated from the *posterior* distribution $\pi(\boldsymbol{\theta}, \lambda^2 | \mathbf{z}^f)$. Then,

$$\mathbf{z}^f_{rep,i} \underset{i.i.d.}{\sim} \mathscr{N}\left(y_{\boldsymbol{\theta}_i}(.), \lambda^2_i\right). \tag{5.19}$$

Figure 5.7 – *Probability distribution of the averaged electric power delivered inside the cell over the time period.*

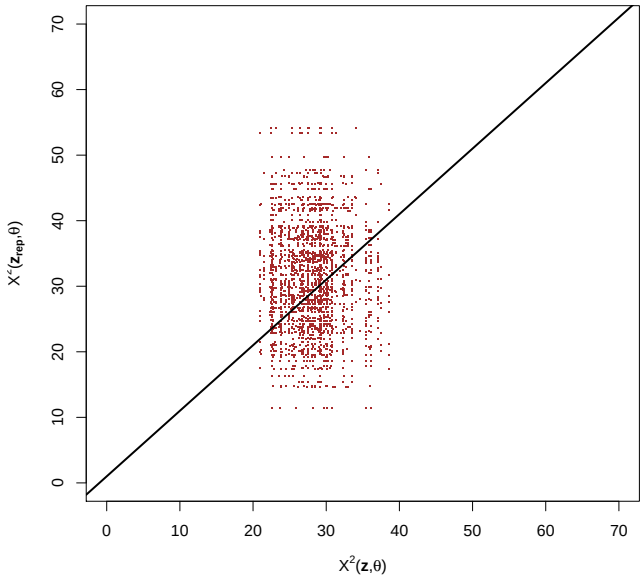


Let us remark that (5.17) is equivalent to the *posterior* mean of the frequentist p-value over the *posterior* distribution of $(\boldsymbol{\theta}, \lambda^2)$. Hence,

$$p_{\mathrm{B}} = \int \mathbb{P}[\mathscr{X}^2_{30} > \mathscr{X}^2(\mathbf{z}^f, \boldsymbol{\theta}, \lambda^2) | (\boldsymbol{\theta}, \lambda^2), \mathbf{z}^f] \pi(\boldsymbol{\theta}, \lambda^2 | \mathbf{z}^f) \mathrm{d}\boldsymbol{\theta} d\lambda^2 \tag{5.20}$$

where $\mathscr{X}^2_{30}$ follows a chi-square distribution with 30 degrees of freedom. Figure 5.8 shows the scatterplot of $\mathscr{X}^2(\mathbf{z}^f_{rep}, \boldsymbol{\theta}, \lambda^2)$ against $\mathscr{X}^2(\mathbf{z}^f, \boldsymbol{\theta}, \lambda^2)$. From this graph, p_{B} is equal to 0.63, that means the model (5.9) is not in question.

Figure 5.8 – *Scatterplot of predictive vs. realized $\mathscr{X}^2$ discrepancies. The posterior p-value is estimated by the proportion of points above the black line.*



5.5 Optimal forecasts of consumption

Bayesian inference is well-suited to assess the epistemic uncertainty of an unknown quantity of interest under the form of a probability distribution. However, in many situations, a point estimate should be required. In this paper, we aim at computing a point estimate of the average electric consumption $\bar{y}_{\boldsymbol{\theta}}$ delivered inside the experimental cell over the time period. The standard approach suggests to compute the maximum (MAP) *a posteriori* of $\bar{y}_{\boldsymbol{\theta}}$. This estimator tends to the maximum likelihood estimator when the number of data tends to infinity. However, such a point estimate ignores the decisional issues, which seems hardly possible in an industrial context. For instance in seismic hazard, the failure probability of a structure should not be underestimated (Damblin et al., 2014). Hence, a smaller weight should be attributed to the under-estimation than the over-estimation. A Bayes estimator expresses the fact that a statistical problem underlies a decision problem by combining two ingredients:

1. the uncertainty affecting the true average power $\bar{y}_{\boldsymbol{\theta}}$ which is quantified by the probability distribution $\pi(\bar{y}_{\boldsymbol{\theta}})$,
2. a utility function $\mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}})$ where d is a forecast of $\bar{y}_{\boldsymbol{\theta}}$. U shall assess the economic consequences for the electric supplier to predict a consumption d to customers instead of $\bar{y}_{\boldsymbol{\theta}}$. U therefore depends on the kind of energy contract to be proposed to customers.

Today, the energy contracts are made according to a description of the building occupied by customers, especially according to the electrical devices inside the building. Eventually, however, these contracts may be modified in light of the effective consumption when it is far away from the initial forecast. In the following, we propose two kinds of new contracts in order to attract the customers.

First proposal The customer has to pay a fixed price per month according to d. In the case where the effective consumption $\bar{y}_{\boldsymbol{\theta}}$ exceeds d, the energy supplier commits to pay

the surplus amount. This is motivated by the guarantee of performance. A company which proposes such a contract should be precisely assessed its benefits by means of a utility function U. Let us assume the energy fee is linear in the electric power. Then,

$$\mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}}) = (b \times d)\mathbf{1}_{\bar{y}_{\boldsymbol{\theta}} > d} + \frac{b \times d}{c(d - \bar{y}_{\boldsymbol{\theta}}) + 1}\mathbf{1}_{\bar{y}_{\boldsymbol{\theta}} \leq d} \tag{5.21}$$

where

— b characterizes the linear relation between the electric power and the electricity price,

— c characterizes the probability $1 - (c(d - \bar{y}_{\boldsymbol{\theta}}) + 1)^{-1}$ that the customer does not purchase the proposal or break it. The higher c, the larger this probability.

In the case where the true consumption $\bar{y}_{\boldsymbol{\theta}}$ is lower than d, the customer can decide that such a proposal is not interesting and does not subscribe to it with probability $1 - (c(d - \bar{y}_{\boldsymbol{\theta}}) + 1)^{-1}$, which should be considered as an indicator of the market price. For this reason, the value of both b and c should be carefully assessed with the help of the trade sector. Given that $\bar{y}_{\boldsymbol{\theta}}$ is uncertain, the optimal consumption forecast, denoted by $\hat{d}$, is the value which maximizes the mean of the expected utility, given by

$$\begin{aligned} \hat{d} &= \operatorname*{argmax}_d \mathbb{E}[\mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}})] \\ &= \operatorname*{argmax}_d \int_{\bar{y}_{\boldsymbol{\theta}}} \mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}}) \pi(\bar{y}_{\boldsymbol{\theta}}) \mathrm{d}\bar{y}_{\boldsymbol{\theta}}. \end{aligned} \tag{5.22}$$

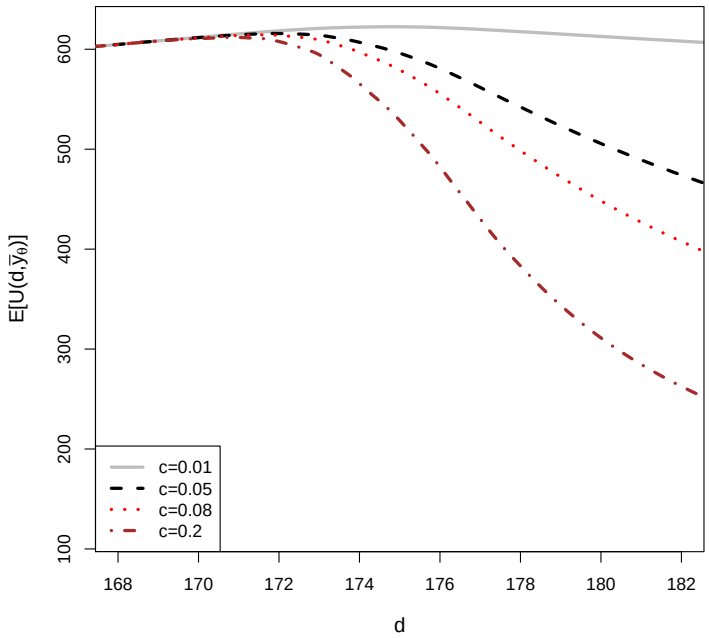
Figure 5.9 shows the results for several values of c, where b has been set to an arbitrary value. We can see the higher c, the lower the estimator $\hat{d}$. The reason is because when c increases, the probability $1-(c(d-\bar{y}_{\boldsymbol{\theta}})+1)^{-1}$ for customers to break the contract increases. Hence, the optimal forecast $\hat{d}$ should not be intuitively too high. Hereafter we propose another utility function which could be used instead of (5.21).

Second proposal A second utility function can be written as

$$\begin{aligned} \mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}}) &= \big(f(d) + cd\big)\mathbf{1}_{\bar{y}_{\boldsymbol{\theta}} > d} + \big(f(d) + c\bar{y}_{\boldsymbol{\theta}}\big)\mathbf{1}_{\bar{y}_{\boldsymbol{\theta}} \leq d} \\ &= f(d) + c \times \min(\bar{y}_{\boldsymbol{\theta}}, d) \end{aligned} \tag{5.23}$$

where $f(d)$ is a decreasing function of d. It is the fixed part of the contract and $c \times \min(\bar{y}_{\boldsymbol{\theta}}, d)$ is the variable part not exceeding d. The optimal decision $\hat{d}$ does a trade off between a large guaranteed amount for the energy supplier but making potentially large losses once the customer's consumption exceeds $\hat{d}$ and a small guaranteed amount but where the customer must most of the time pay its actual consumption.

Figure 5.9 – *The expectation* $\mathbb{E}[\mathrm{U}(d, \bar{y}_{\boldsymbol{\theta}})]$ *corresponding to the first proposal is plotted along the vertical axis against the value of* d *along the horizontal axis.*



5.6 Conclusion

Understanding the pattern of thermal behavior in buildings is conducted with the help of computer codes. In this paper, a dynamic thermal code implemented with the *Dymola* software has been used to compute predictions of the electric power injected inside an experimental cell over a time period. From the need of code validation, we have conducted a framework to assess their uncertainty.

Most of the time, the lack of accuracy (that is a large uncertainty in the predictions) may arise from many causes such as a bad implementation of the source code, a poor mathematical representation or even a bad specification of the code inputs. In this paper, we have focused on the impact of the epistemic uncertainty tainting the vector of the unknown code parameters within a Bayesian setting. This calibration stage has been followed by a validation stage where this uncertainty has been propagated to the code outputs on the time period, then to the average consumption on the time period. Finally, some optimal consumption estimators have been calculated based on an appropriate utility function. Note that the simulations of the Dymola code were fast to be returned (a few seconds). It they were not, an emulator such as a Gaussian process or a polynomial chaos expansion should have been constructed in the calibration stage, leading to an additional source of uncertainty to be taken into account.

In this study, some utility functions have been specified to commit for an overall consumption forecast to customers according to a new kind of energy contract. The way to construct them is a considerable challenge, which should be tackled with the help of the trade sector of the energy supplier. The functions proposed by us should in fact be viewed as a guide for building more realistic ones. Another challenging issue is how to model the albedo. In the paper, this uncertain parameter is assumed constant but it might be more realistic to model it by a temporal series. That would thus require more sophisticated

calibration procedures.

At this stage, our research has been explanatory to put in light the steps to follow. Then, we would like to deal with data collected in a real building. Besides, we might question whether it is possible to predict thermal quantities for future time periods with a thermal code calibrated on the past.

Bibliography

AIAA (1998). Guide for the verification and validation of computational fluid dynamics simulations. *Reston VA: American Institute of Aeronautics and Astronautics*, AIAA-G-077-1998. 107

Bayarri, M. J., Berger, J. O., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., and Tu, J. (2007). A framework for validation of computer models. *Technometrics*, 49:138–154. 107, 111

Bernardo, J. M. and Smith, A. F. M. (1994). *Bayesian Theory*. Wiley, London, 1 edition. 103, 104

Bontemps, S., Kaemmerlen, A., Blatman, G., and Mora, L. (2013). Reliability of dynamic simulation models for building energy in the context of low energy buildings. *13th Conference of International Building Performance Simulation Association, Chambéry, France, August 26-28*. 103, 104, 105, 106

Campbell, K. (2006). Statistical calibrations of computer simulations. *Reliability Engineering and System Safety*, 91(10-11):1358–1363. 103

Damblin, G., Keller, M., Pasanisi, A., and Parent, E. (2014). Approche décisionnelle bayésienne pour estimer une courbe de fragilité. *Journal de la Société Française de Statistique*, 155(3). 113

Gelman, A., Meng, X., and Stern, H. (1996). Posterior predictive assessment of model fitness via realized discrepancies. *Statistica Sinica*, 6(4):733–807. 111

Kennedy, M. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B, Methodological*, 63(3):425–464. 103, 105, 106, 110

Loeppky, D., Bingham, D., and Welch, W. (2006). Computer model calibration or tuning in practice. Technical report. 105

Meng, X. (1994). Posterior predictive p-values. *The Annals of Statistics*, 3(3):1142–1160. 111

Pasanisi, A., Keller, M., and Parent, E. (2012). Estimation of a quantity of interest in uncertainty analysis: Some help from bayesian decision theory. *Reliability Engineering and System Safety*, 100:93–101. 103, 107

Roache, P. (1998). Verification of codes and calculations. *AIAA Journal*, 36:696–702. 107

Robert, C. and Casella, G. (1998). *Monte Carlo Statistical Methods*. Springer-Verlag. 109

Roy, C. and Oberkampf, W. (2011). A comprehensive framework for verification, validation and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25-28):2131–2144. 103, 107

Saltelli, A., Chan, K., and Scott, E. (2000). *Sensitivity Analysis*. Wiley, New York. 106

Spitz, C. (2012). *Analyse de la fiabilité des outils de simulation et des incertitudes de métrologie appliquée à l'efficacité énergétique des bâtiments*. PhD thesis, Université de Grenoble. 103

Ulmo, J. and Bernier, J. (1973). *Eléments de Décision Statistique*. PUF. 104

Wang, S., Chen, W., and Tsui, K. (2009). Bayesian validation of computer models. *Technometrics*, 51(4):439–451. 107

Chapitre 6

Conclusions et perspectives

Dans ce mémoire, la validation des codes de calcul est abordée comme la démarche statistique visant à quantifier l'incertitude induite par la prédiction d'un système physique à l'aide des réponses d'un code de calcul l'approchant, et en tenant compte du contexte industriel d'utilisation du code. Cette question est fondamentale, tant en pratique, la conception d'un code de calcul ne s'accompagne que rarement de protocoles bien établis visant à s'assurer qu'il peut être utilisé avec un degré de confiance suffisant comme complément ou moyen de substitution à l'expérience réelle.

Nos développements ont trait aux méthodes de calage et de validation des codes de calcul boîtes noires qui s'appuient sur un modèle statistique reliant les réponses du code aux mesures physiques disponibles. Lorsque les simulations issues du code sont coûteuses en temps d'exécution, ce modèle est estimé dans un cadre bayésien grâce à des techniques modulaires permettant de séparer l'estimation des paramètres incertains du code, de l'estimation des hyperparamètres du processus gaussien (Liu et al., 2009). Le processus gaussien, construit à partir d'un nombre réduit de simulations, fait office d'émulateur du code capable de traduire l'incertitude sur ses réponses dans un domaine d'intérêt des variables d'entrée (Sacks et al., 1989). Une fois le modèle estimé, des intervalles de crédibilité quantifiant l'incertitude de prédiction du système physique peuvent être calculés. Cette démarche statistique fournit une réponse quantitative à la première étape du problème de validation d'un code de calcul, appelée approche prédictive (dont font l'objet les chapitres 2, 3 et 4). La deuxième étape consiste à déterminer si cette incertitude est acceptable vis à vis d'une certaine problématique industrielle (dont fait l'objet le chapitre 5).

Le modèle statistique utilisé prend la forme d'une équation reliant les réponses du code aux mesures physiques, en intercalant éventuellement entre les deux une fonction appelée « erreur de code » qui traduit l'inadéquation éventuelle qu'il peut exister entre la réponse du code et le système physique modélisé. Cette fonction, souvent inconnue, est aussi approchée par un processus gaussien dans les méthodes que nous avons étudiées. A ce stade, nous aurions pu formuler une définition de la validation selon laquelle un code est jugé valide uniquement si cette erreur de code est suffisamment peu probable au regard des résultats de la procédure de test que nous proposons dans la chapitre 3. Autrement dit, nous réfuterions le calcul de l'incertitude de prédiction d'un système physique à partir de prédictions numériques corrigées par une erreur de code. Cependant, nous croyons qu'une telle définition est trop restrictive, en particulier lorsque l'architecture interne du code de calcul ne peut pas être modifiée facilement.

L'erreur de code traduit le fait qu'*a priori*, le code de calcul n'est pas capable de reproduire exactement le système physique. La justification avancée par Kennedy et O'Ha-

gan (2001) est que le code n'est conçu qu'à partir d'une modélisation mathématique imparfaite de la réalité physique sous-jacente. En nous inspirant des travaux déjà existants, nous avons illustré, d'une part, les problèmes d'identifiabilité statistique du modèle (2.42) incluant l'erreur de code, et d'autre part, la construction des intervalles de crédibilité à partir des réponses du code corrigées par l'erreur de code. Pour pouvoir justifier (ou réfuter) l'utilisation de telles prédictions, il nous a semblé pertinent de construire une procédure de test statistique mettant en compétition l'équation reliant directement les réponses du code au système physique, avec celle plus complexe qui inclut cette erreur de code. Pour cela, nous avons mis en place une technique bayésienne de sélection de modèle qui consiste en une version modifiée du facteur de Bayes, appelée facteur de Bayes intrinsèque, introduit par Berger et Pericchi (1996). La facteur de Bayes intrinsèque nous permet d'utiliser des lois *a priori* impropres sur le paramètre du code et le paramètre d'échelle dans chacun des deux modèles et d'utiliser une loi *a priori* objective sur la longueur de corrélation du processus gaussien modélisant l'erreur de code. Dans le cas où le résultat de la procédure de test ne serait pas assez décisif en faveur de l'un ou l'autre des modèles, l'incertitude de prédiction du système physique pourrait être obtenue en effectuant la moyenne des incertitudes de prédiction calculées pour chacun des deux modèles, pondérée par la probabilité *a posteriori* de chaque modèle. On parle dans ce cas d'agrégation de modèles (Bayarri et al., 2002; Hoeting et al., 1999).

Les calculs du facteur de Bayes intrinsèque ont été menés sous l'hypothèse que le code de calcul est linéaire en ses paramètres, ce qui réduit significativement leur complexité. Pour des codes non linéaires pour lesquels il n'existe pas de valeur de référence permettant d'envisager la linéarisation du code autour de celle-ci, des méthodes numériques avancées devront être utilisées pour calculer avec précision les vraisemblances marginales de chacun des deux modèles en compétition.

Un deuxième axe de travail important en validation des codes de calcul est celui du plan d'expériences numérique utilisé pour construire l'émulateur de type processus gaussien. Dans la littérature, beaucoup de travaux étudient les plans d'expériences numériques qui occupent de manière optimale l'espace des variables d'entrée au sens de différents critères géométriques et/ou statistiques. Pour la validation des codes de calcul coûteux nécessitant une étape de calage (autrement dit pour les codes entachés d'une incertitude paramétrique), ces plans d'expériences ne sont pas les mieux adaptés (voir section 2.3). C'est pourquoi, nous avons proposé dans le chapitre 4 de nouveaux algorithmes séquentiels de construction de plans d'expériences numériques en utilisant le critère de l'amélioration espérée (Jones et al., 1998). Nous montrons que ces plans permettent de réduire l'erreur de calage au sens de la distance de Kullback-Leibler lorsqu'un émulateur processus gaussien remplace le code dans la procédure d'estimation. Une autre piste de travail pourrait être de construire des plans d'expériences séquentiels à la fois numériques et physiques [1] permettant de réduire simultanément l'erreur de calage et l'incertitude de prédiction du système physique. En l'absence d'incertitude paramétrique, les plans séquentiels proposés par Le Gratiet et Cannamela (2015) pour l'émulation multi-fidélité pourraient être utilisés pour réduire au mieux l'incertitude de prédiction. En présence d'incertitude paramétrique, une piste de recherche pourrait être de construire des plans d'expériences séquentiels assurant un compromis, dans un sens à définir, entre ces deux objectifs.

Nous avons consacré la dernière partie du mémoire à l'application d'une méthode de calage et de validation à un code de calcul simulant le comportement thermique d'une cellule expérimentale. Après avoir effectué le calage du code à partir des mesures de puis-

1. qui consisteraient à déterminer quelles sont les expériences physiques à effectuer.

sance électrique relevées sur une période de temps de 7 jours, nous avons quantifié l'incertitude affectant la puissance thermique injectée à l'intérieur de la cellule en fonction du temps, puis l'incertitude affectant la puissance moyenne injectée sur la période. Enfin, grâce à la théorie de la décision bayésienne, nous avons proposé des estimateurs non dominés de consommation moyenne pour la construction d'une offre de garantie de performance. Le calcul de ces estimateurs s'appuie, d'une part, sur l'incertitude de prédiction de la puissance moyenne provenant de la validation du code, et d'autre part, sur le choix d'une fonction de coût qui doit évaluer les pertes et bénéfices associées au choix d'un estimateur ponctuel.

Des perspectives découlent de ces premières contributions. En particulier, les travaux actuels visant à réduire l'incertitude d'approximation d'un code de calcul par un émulateur processus gaussien devraient pouvoir bénéficier à la problématique de la validation. Un autre enjeu important est celui de la gestion de la taille des matrices utilisées dans les procédures d'estimation, lorsque le nombre de simulation utilisé pour construire le processus gaussien est important. Pour cela, Gramacy et Apley (2015) construisent un émulateur processus gaussien local à partir d'une base d'apprentissage de taille réduite, sans toutefois dégrader la prédiction que l'on aurait pu obtenir en utilisant toutes les simulations disponibles. Enfin, la grande dimension des paramètres peut rendre l'estimation du modèle (2.42) trop coûteuse à mettre en œuvre. Là encore, les avancées concernant l'échantillonnage MCMC devraient pouvoir bénéficier aux méthodes de validation. Des alternatives sont aussi possibles en utilisant les méthodes issues de l'analyse bayésienne linéaire et de l'assimilation de données, qui reposent sur des calculs plus simples et réalisables en grande dimension.

Bibliographie

Bayarri, M. J., Berger, J. O., Higdon, D., Kennedy, M., Kottas, A., Paulo, R., Sacks, P. R., Cafeo, J. A., Cavendish, J., Lin, C.-H., et Tu, J. (2002). A framework for validation of computer models. *Tech. Rep. 128, National Institute of Statistical Sciences*. 120

Berger, J. et Pericchi, L. (1996). The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433) :109–122. 120

Gramacy, R. et Apley, D. (2015). Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2) :561–578. 121

Hoeting, J., Madigan, D., Raftery, A., et Volinsky, C. (1999). Bayesian model averaging : A tutorial. *Statistical Science*, 14(4) :382–417. 120

Jones, D., Schonlau, M., et Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4) :455–492. 120

Kennedy, M. et O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society, Series B, Methodological*, 63(3) :425–464. 119

Le Gratiet, L. et Cannamela, C. (2015). Cokriging-based sequential design strategies using fast cross-validation techniques for multi-fidelity computer codes. *Technometrics*, 57(3) :418–427. 120

Liu, F., Bayarri, M., et Berger, J. (2009). Modularization in bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1) :119–150. 119

Sacks, J., W.J., W., Mitchell, T., et Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4) :409–423. 119

Annexe A

Le processus aléatoire gaussien

Nous rappelons d'abord le cadre général de l'étude des processus aléatoires, puis nous développons les équations du processus gaussien lorsque celui-ci est utilisé pour prédire une fonction boîte noire coûteuse. Soit $(\Omega, \mathscr{F}, \mathbb{P})$ un espace probabilisé dans lequel Ω désigne l'ensemble des événements élémentaires d'une expérience aléatoire, $\mathscr{F}$ une tribu sur Ω, et $\mathbb{P}$ une mesure de probabilité sur Ω. De plus, nous notons $\mathcal{V} \subset \mathbb{R}^d$ l'espace d'indexation.

A.1 Définition d'un processus aléatoire

Definition A.1.0.1 ***(Variable aléatoire réelle)*** *Une variable aléatoire réelle sur* $(\Omega, \mathscr{F}, \mathbb{P})$ *est une fonction mesurable* $\mathrm{Y} : (\Omega, \mathscr{F}) \longrightarrow (\mathbb{R}, \mathscr{B}(\mathbb{R}))$.

Definition A.1.0.2 ***(Processus aléatoire à valeurs réelles)*** *Un processus aléatoire à valeurs réelles* $\{\mathrm{Y}_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ *désigne une famille de variables aléatoires réelles définies sur le même espace de probabilité* $(\Omega, \mathscr{F}, \mathbb{P})$.

Lorsque l'indice $\mathbf{v} \in \mathcal{V}$ est fixé, $\mathrm{Y}_{\mathbf{v}} : \Omega \longrightarrow \mathbb{R}$ est une variable aléatoire. Lorsque l'événement élémentaire $\omega \in \Omega$ est fixé, la fonction $\mathbf{v} \longrightarrow \mathrm{Y}_{\mathbf{v}}(\omega)$ représente une trajectoire du processus.

Definition A.1.0.3 ***(Lois fini-dimensionnelles)*** *Considérons n indices,* $\mathbf{v}_1 \in \mathcal{V}, \cdots, \mathbf{v}_n \in \mathcal{V}$. *La loi de probabilité du vecteur aléatoire* $\mathrm{Y}_{\mathbf{v}_1,\cdots,\mathbf{v}_n} := (\mathrm{Y}_{\mathbf{v}_1}, \cdots, \mathrm{Y}_{\mathbf{v}_n})$ *est appelée loi fini-dimensionnelle du processus aléatoire* $\{\mathrm{Y}_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$.

La loi de probabilité d'un processus aléatoire est complètement déterminée par ses lois fini-dimensionnelles. Inversement, le théorème de Kolmogorov nous assure l'existence d'un processus aléatoire construit en spécifiant les lois de probabilité fini-dimensionnelles des vecteurs aléatoires $(\mathrm{Y}_{\mathbf{v}_1}, \cdots, \mathrm{Y}_{\mathbf{v}_n})$ prescrites pour tout vecteur $(\mathbf{v}_1, \cdots, \mathbf{v}_n) \in \mathcal{V}^n$, sous des hypothèses de symétrie et de compatibilité de celles-ci. Sous réserve que $\mathbb{E}[\mathrm{Y}_{\mathbf{v}}^2] < \infty$ pour tout $\mathbf{v} \in \mathcal{V}$, la fonction de covariance d'un processus aléatoire $\{\mathrm{Y}_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ peut s'écrire comme une fonction k des indices $\mathbf{v}$ et $\mathbf{v}'$:

$$Cov(\mathrm{Y}_{\mathbf{v}}, \mathrm{Y}_{\mathbf{v}'}) := k(\mathbf{v}, \mathbf{v}').$$

La covariance $\mathrm{C}(.,.)$ est une forme bilinéaire symétrique semi-définie positive.

Definition A.1.0.4 ***(Processus aléatoire stationnaire)*** *Supposons que* $\mathcal{V} = \mathbb{R}^d$. *Un processus aléatoire est stationnaire si pour tout* $\mathbf{v}_1, \cdots, \mathbf{v}_n \in \mathcal{V}$ *et pour tout* $\mathbf{h} \in \mathbb{R}^d$, *alors* $\mathrm{Y}_{\mathbf{v}_1,\cdots,\mathbf{v}_n}$ *a même loi que* $\mathrm{Y}_{\mathbf{v_1}+\mathbf{h},\cdots,\mathbf{v_n}+\mathbf{h}}$.

La propriété de stationnarité d'un processus aléatoire vérifiant $\mathbb{E}[Y_{\mathbf{v}}^2] < \infty$ pour tout $\mathbf{v} \in \mathcal{V}$ implique que $\{Y_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ ont même moyenne et que la fonction covariance s'écrit comme une fonction de $\mathbf{h} = \mathbf{v} - \mathbf{v}'$: $k(\mathbf{h}) = k(\mathbf{v}, \mathbf{v}')$. Dans le cas particulier des processus gaussiens, il y a équivalence.

L'ergodicité d'un processus aléatoire est une propriété statistique permettant son estimation à partir d'une seule de ses réalisations. Dans le cas particulier d'un processus gaussien, la propriété d'ergodicité peut être reliée à la propriété de stationnarité.

Theorème A.1.0.1 *(**Ergodicité d'un processus gaussien** (Cressie, 1993)) Un processus gaussien stationnaire est ergodique si* $\lim_{||\mathbf{h}||\to\infty} k(\mathbf{h}) = 0$.

Nous donnons deux définitions relatives à la continuité d'un processus aléatoire.

Definition A.1.0.5 *(**Continuité en moyenne quadratique**) Un processus aléatoire* $\{Y_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ *vérifiant* $\mathbb{E}[Y_{\mathbf{v}}^2] < \infty$ *pour tout* $\mathbf{v} \in \mathcal{V}$ *est continue en moyenne quadratique sur* $\mathcal{V}$ *si pour tout* $\mathbf{v}_0 \in \mathcal{V}$

$$\lim_{\mathbf{v}\to\mathbf{v}_0} \mathbb{E}\big[\big(Y(\mathbf{v}) - Y(\mathbf{v}_0)\big)^2\big] = 0.$$

La propriété de continuité en moyenne quadratique d'un processus stationnaire peut être reliée à la continuité de la fonction de covariance du processus via la proposition suivante.

Proposition A.1.0.1 *(Abrahamsen, 1997) Un processus aléatoire stationnaire* $\{Y_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ *sur* $\mathcal{V} = \mathbb{R}^d$ *est continu en moyenne quadratique si et seulement si sa fonction de covariance* $k(.)$ *est continue en* 0.

Definition A.1.0.6 *(**Continuité presque sûre des trajectoires**) Un processus aléatoire* $\{Y_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ *possède des trajectoires continues presque sûrement si*

$$\mathbb{P}[\omega \in \Omega\,;\ \forall \mathbf{v}_0 \lim_{\mathbf{v}\to\mathbf{v}_0} Y(\omega, \mathbf{v}) - Y(\omega, \mathbf{v}_0) = 0] = 1.$$

La continuité presque sûre d'un processus n'implique pas sa continuité en moyenne quadratique et inversement. Pour une présentation plus détaillée des propriétés de continuité et de dérivabilité des processus aléatoires, utiles pour l'émulation d'un code de calcul (voir section A.3), nous renvoyons à l'ouvrage de Santner et al. (2003) et à la thèse de Bettinger (2009).

A.2 Vecteur et processus gaussien

Definition A.2.0.1 *(**Vecteur aléatoire gaussien**) Un vecteur aléatoire* $\mathbf{Y} := (Y_1, \cdots, Y_n) \in \mathbb{R}^n$ *est gaussien si pour tout* $\lambda_1, \cdots, \lambda_n \in \mathbb{R}$, *la variable aléatoire* $\sum_{i=1}^n \lambda_i Y_i$ *est gaussienne.*

La loi d'un vecteur gaussien $\mathbf{Y}$ de dimension n est complètement déterminée par ses deux premiers moments :

$$\mathbf{m} = (\mathbb{E}[Y(\mathbf{v}_1)], \cdots, \mathbb{E}[Y(\mathbf{v}_n)])^{\mathrm{T}}, \tag{A.1}$$

qui est le vecteur moyenne de Y et

$$\mathrm{V} = ((cov(Y_i, Y_j))_{1\le i,j\le n}, \tag{A.2}$$

qui est la matrice de variance-covariance de $\mathbf{Y}$. Lorsque V est définie positive, $\mathbf{Y}$ admet une densité de probabilité

$$f(\mathbf{y}) = \frac{|\mathrm{V}|^{-1/2}}{(2\pi)^{n/2}} \exp\Big(-\frac{1}{2}(\mathbf{y}-\mathbf{m})^{\mathrm{T}}\mathrm{V}^{-1}(\mathbf{y}-\mathbf{m})\Big). \tag{A.3}$$

Proposition A.2.0.1 *Soit* $\mathbf{Y}_1, \mathbf{Y}_2$ *deux vecteurs gaussien tels que*

$$\begin{pmatrix}\mathbf{Y}_1\\ \mathbf{Y}_2\end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{m_1}\\ \mathbf{m_2}\end{pmatrix}, \begin{pmatrix}\mathrm{V}_{1,1} & \mathrm{V}_{1,2}\\ \mathrm{V}_{2,1} & \mathrm{V}_{2,2}\end{pmatrix}\right). \tag{A.4}$$

Supposons que $\mathrm{V}_{1,1}$ *est inversible. Alors la loi conditionnelle* $\mathrm{Y}_2|\mathrm{Y}_1$ *est gaussienne*

$$\mathcal{N}\left(\mathbf{m_2} + \mathrm{V}_{2,1}\mathrm{V}_{1,1}^{-1}(\mathbf{Y}_1 - \mathbf{m_1}), \mathrm{V}_{2,2} - \mathrm{V}_{2,1}\mathrm{V}_{1,1}^{-1}\mathrm{V}_{1,2}\right). \tag{A.5}$$

L'expression analytique gaussienne de la loi de probabilité d'un premier vecteur gaussien $\mathbf{Y}_1$ conditionnellement à un second $\mathbf{Y}_2$ permet d'obtenir une expression analytique de l'émulateur d'un code de calcul de type processus gaussien (voir section A.3).

Definition A.2.0.2 *(**Processus gaussien**) Un processus aléatoire* $\{\mathbf{Y_v}\}_{\mathbf{v}\in\mathcal{V}}$ *est gaussien si chacune de ses lois fini-dimensionnelles* $\mathrm{Y}_{\mathbf{v}_1,\cdots,\mathbf{v}_n} := (\mathrm{Y}_{\mathbf{v}_1}, \cdots, \mathrm{Y}_{\mathbf{v}_n})$ *est un vecteur gaussien.*

Considérons,

$$m : \mathbf{v} \in \mathcal{V} \longleftrightarrow m(\mathbf{v}) = \mathbb{E}[\mathrm{Y}_{\mathbf{v}}] \in \mathbb{R}^q \tag{A.6}$$

une fonction moyenne et

$$\mathrm{K} : (\mathbf{v}, \mathbf{v}') \in \mathcal{V} \times \mathcal{V} \longleftrightarrow \mathrm{K}(\mathbf{v}, \mathbf{v}') = corr(\mathrm{Y}_{\mathbf{v}}, \mathrm{Y}_{\mathbf{v}'}) \tag{A.7}$$

une fonction de corrélation [1]. La donnée de $m(.)$, d'un paramètre d'échelle σ^2 et de $\mathrm{K}(.,.)$ semi-définie positive entraîne l'existence et l'unicité d'un processus aléatoire gaussien de lois fini-dimensionnelles induites par ces trois quantités. Il est noté dans ce mémoire

$$\mathrm{Y}(.) \sim \mathcal{PG}(m(.), \sigma^2\mathrm{K}(.,.)). \tag{A.8}$$

Les propriétés des processus gaussiens découlent donc des propriétés des vecteurs gaussiens.

Proposition A.2.0.2 *Un processus gaussien* $\{\mathrm{Y}_{\mathbf{v}}\}_{\mathbf{v}\in\mathcal{V}}$ *est stationnaire si et seulement si sa fonction moyenne* $m(\mathbf{v})$ *est constante et sa fonction de covariance* $\sigma^2\mathrm{K}(\mathbf{v}, \mathbf{v}')$ *dépend uniquement de* $\mathbf{h} = \mathbf{v} - \mathbf{v}'$.

En pratique, lorsque l'on veut modéliser la réponse d'un code de calcul comme la réalisation d'un processus aléatoire (voir section A.3), on dispose seulement d'une réalisation partielle du processus

$$\{\mathrm{Y}_{\mathbf{v}_1}(\omega), \cdots, \mathrm{Y}_{\mathbf{v}_n}(\omega)\},$$

ce qui nous éloigne du cadre classique dans lequel nous disposerions d'un ensemble de trajectoires $\{\mathrm{Y}_{(.)}(\omega_1), \cdots, \mathrm{Y}_{(.)}(\omega_n)\}$. C'est la propriété d'ergodicité qui justifie la validité de l'inférence lorsque le processus gaussien et stationnaire et lorsque $\lim_{||\mathbf{h}||\to\infty} \mathrm{K}(\mathbf{h}) = 0$ (voir théorème A.1.0.1).

Proposition A.2.0.3 *(**Produit de deux densités gaussiennes**) Le produit de deux densités de probabilité gaussiennes* $f(y|\mathbf{m}_1, \mathrm{V}_1)$ *et* $f(y|\mathbf{m}_2, \mathrm{V}_2)$ *est une densité de probabilité gaussienne à une constante de normalisation près :*

$$f(y|\mathbf{m}_1, \mathrm{V}_1) f(y|\mathbf{m}_2, \mathrm{V}_2) = \mathrm{C}^{-1} f(y|\mathbf{m}_3, \mathrm{V}_3) \tag{A.9}$$

où $\mathbf{m}_3 = \mathrm{V}_3(\mathrm{V}_1^{-1}\mathbf{m}_1 + \mathrm{V}_2^{-1}\mathbf{m}_2)$ *et* $\mathrm{V}_3 = (\mathrm{V}_1^{-1} + \mathrm{V}_2^{-1})^{-1}$. *De plus,*

$$\mathrm{C}^{-1} = (2\pi)^{-n/2}|\mathrm{V}_1 + \mathrm{V}_2|^{-1/2} \exp -\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^{\mathrm{T}}(\mathrm{V}_1 + \mathrm{V}_2)^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

est également une densité gaussienne en $\mathbf{m}_1$ *ou* $\mathbf{m}_2$.

1. aussi appelée noyau de corrélation

A.3 Modélisation par processus gaussien

Soit $f : \mathbf{v} \in \mathcal{V} \subset \mathbb{R}^d \longrightarrow f(\mathbf{v}) \in \mathbb{R}$ une fonction déterministe à valeurs réelles. Le processus gaussien est un outil de modélisation statistique permettant d'approcher la réponse $f(\mathbf{v})$ pour tout $\mathbf{v} \in \mathcal{V}$ à partir d'une base d'apprentissage constituée d'un nombre réduit d'évaluations de la fonction. À travers ce mémoire, la modélisation par processus gaussien est utilisée pour approcher la réponse $f(\mathbf{v})$ lorsque celle-ci est :

1. difficile à obtenir, typiquement lorsque f est un code de calcul dont les simulations sont coûteuses en temps d'exécution. On parle alors d'émulation par processus gaussien.
2. impossible à obtenir, typiquement lorsque f est une fonction inconnue (l'erreur de code). Dans ce cas, on parle soit d'émulation par processus gaussien (vision bayésienne), soit de régression par processus gaussien (voir section 2.4.5).

En adoptant une vision bayésienne, le processus gaussien constitue un *a priori* fonctionnel sur la réponse de la fonction f que l'on cherche à prédire (Currin et al., 1991) :

$$f(.) \sim \mathcal{PG}(h(.)^{\mathrm{T}}\boldsymbol{\beta}_f, \sigma_f^2 \mathrm{K}_{\boldsymbol{\Psi}_f}(.,.)), \tag{A.10}$$

où $\boldsymbol{\beta}_f$, σ_f^2 et $\boldsymbol{\Psi}_f$ sont les hyperparamètres spécifiant l'espérance et la structure de variance-covariance du processus et $h(\mathbf{v}) = \left(h_1(\mathbf{v}), \cdots, h_p(\mathbf{v})\right)^{\mathrm{T}}$ est un vecteur de fonctions de régression. Soient $\mathbf{v} \in \mathcal{V}$ et $\mathbf{v}' \in \mathcal{V}$, alors

$$Cov(f(\mathbf{v}), f(\mathbf{v}')) = \sigma_f^2 \mathrm{K}_{\boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}'). \tag{A.11}$$

Supposons que la fonction f soit évaluée en un nombre M de configurations d'entrées

$$\mathbf{D}_{\mathrm{M}} = \{\mathbf{v}_1^{\mathrm{D}}, \cdots, \mathbf{v}_{\mathrm{M}}^{\mathrm{D}}\},$$

appelé plan d'expériences numériques (voir annexe A.4.1) et notons $f(\mathbf{D}_{\mathrm{M}})$ les M réponses correspondantes

$$f(\mathbf{D}_{\mathrm{M}}) = \left(f(\mathbf{v}_1^{\mathrm{D}}), \cdots, f(\mathbf{v}_{\mathrm{M}}^{\mathrm{D}})\right)^{\mathrm{T}}.$$

La matrice de corrélation de $f(\mathbf{D}_{\mathrm{M}})$ induite par la fonction de corrélation $\mathrm{K}_{\boldsymbol{\Psi}_f}(.,.)$ s'écrit

$$\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_{\mathrm{M}}) := \Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_{\mathrm{M}}, \mathbf{D}_{\mathrm{M}})$$

où

$$\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_{\mathrm{M}})(i,j) = \mathrm{K}(\mathbf{v}_i, \mathbf{v}_j). \tag{A.12}$$

La distribution du processus *a posteriori* est donnée par

$$f(.)|f(\mathbf{D}_{\mathrm{M}}) \sim \mathcal{PG}(\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(.), \mathrm{V}_{\sigma_f^2, \boldsymbol{\Psi}_f}(.)). \tag{A.13}$$

La moyenne *a posteriori* $\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(.)$ s'écrit

$$\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v}) = h(\mathbf{v})^{\mathrm{T}}\boldsymbol{\beta}_f + \Sigma_{\boldsymbol{\Psi}_f}^{\mathrm{T}}(\mathbf{v}, \mathbf{D}_{\mathrm{M}}) \Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_{\mathrm{M}})^{-1} \left(f(\mathbf{D}_{\mathrm{M}}) - h(\mathbf{D}_{\mathrm{M}})^{\mathrm{T}}\boldsymbol{\beta}_f\right) \tag{A.14}$$

et la variance *a posteriori* $\mathrm{V}_{\sigma_f^2, \boldsymbol{\Psi}_f}(.)$ s'écrit

$$\mathrm{V}_{\sigma_f^2, \boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}') = \sigma_f^2 \left(\mathrm{K}_{\boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}') - \Sigma_{\boldsymbol{\Psi}_f}^{\mathrm{T}}(\mathbf{v}, \mathbf{D}_{\mathrm{M}}) \Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_{\mathrm{M}})^{-1} \Sigma_{\boldsymbol{\Psi}_f}(\mathbf{v}', \mathbf{D}_{\mathrm{M}})\right). \tag{A.15}$$

où le vecteur $\Sigma^{\mathrm{T}}_{\boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{D}_{\mathrm{M}})$ a pour j-ème cordonnée $\mathrm{K}(\mathbf{v}, \mathbf{v}_j)$.

Le processus gaussien (A.13) est le processus conditionné aux réponses $f(\mathbf{D}_{\mathrm{M}})$. Il quantifie l'incertitude *a posteriori* sur la valeur de $f(\mathbf{v})$ pour tout $\mathbf{v} \in \mathcal{V}$ sous la forme d'une distribution gaussienne

$$f(\mathbf{v})|f(\mathbf{D}_{\mathrm{M}}) \sim \mathcal{N}\left(\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v}), \mathrm{V}_{\sigma^2_f, \boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v})\right). \tag{A.16}$$

Sa moyenne est appelée BLUP (*Best Linear Unbiaised Predictor*) qui est le prédicteur linéaire sans biais $\hat{f}$ de f qui minimise l'écart quadratique moyen

$$\mathbb{V}[f(\mathbf{v}) - \hat{f}(\mathbf{v})] = \mathbb{E}[(f(\mathbf{v}) - \hat{f}(\mathbf{v}))^2]. \tag{A.17}$$

Le BLUP interpole les réponses d'apprentissages. Donc pour $1 \leq i \leq \mathrm{M}$:

$$\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v}^{\mathrm{D}}_i) = f(\mathbf{v}^{\mathrm{D}}_i)$$

et

$$\mathrm{V}_{\sigma^2_f, \boldsymbol{\Psi}_f}(\mathbf{v}^{\mathrm{D}}_i, \mathbf{v}^{\mathrm{D}}_i) = 0.$$

Ces deux propriétés étaient attendues étant donnée la nature déterministe de f.

Le choix de $\mathrm{K}_{\boldsymbol{\Psi}_f}(.)$ dépend de la régularité de la fonction f. Nous rappelons ci-dessous les fonctions de covariances les plus souvent utilisées lorsque f est un code de calcul boîte noire.

Les fonctions de corrélations exponentielles Soit $\boldsymbol{\Psi}_f := (\boldsymbol{\theta}, \nu)$ où $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d) > 0$ et $0 < \nu \leq 2$:

$$\mathrm{K}_{\boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}') = \prod_{j=1}^{d} \exp - \frac{|v_j - v'_j|^{\nu}}{\theta_j} \tag{A.18}$$

Ce type de fonction est construit comme un produit de fonctions de corrélation unidimensionnelles. Chaque paramètre θ_j spécifie la longueur de corrélation (la portée) dans la direction j. Un processus gaussien basé sur (A.18) est continu en moyenne quadratique et ses trajectoires sont presque sûrement continues. La régularité du processus dépend de la valeur du paramètre ν. Lorsque $\nu = 2$, le noyau est gaussien et ses trajectoires sont C^{∞}.

Les fonctions de corrélation de Matérn Soit $\boldsymbol{\Psi}_f := (\boldsymbol{\theta}, \nu)$ où $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d) > 0$ et $\nu > -1$:

$$\mathrm{K}_{\boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}') = \prod_{j=1}^{d} \frac{(\theta_j |v_j - v'_j|)^{\nu}}{\Gamma(\nu) 2^{\nu-1}} \mathrm{J}_{\nu}(\theta_j |v_j - v'_j|) \tag{A.19}$$

Comme précédemment, la régularité du processus est déterminée par la valeur de l'hyperparamètre ν tandis que $\boldsymbol{\theta}$ règle les longueurs de corrélation dans chacune des directions.

Les fonctions de corrélation exponentielles et de Matérn définissent des processus stationnaires car la corrélation entre deux réponses $f(\mathbf{v})$ et $f(\mathbf{v}')$ ne dépend que des différences $h_j = |v_j - v'_j|$. Il est également possible de spécifier une fonction de corrélation qui ne dépend que de la norme $||\mathbf{v} - \mathbf{v}'||$, appelée fonction de covariance isotrope. Dans le chapitre 3, nous utilisons une fonction de corrélation gaussienne isotrope :

$$\mathrm{K}_{\theta}(\mathbf{v}, \mathbf{v}') = \exp - \frac{||\mathbf{v} - \mathbf{v}'||^2}{\theta}. \tag{A.20}$$

L'estimation des hyperparamètres En pratique, $\boldsymbol{\beta}_f$, $\boldsymbol{\Psi}_f$ et σ_f^2 ne sont pas connus et doivent donc être estimés à partir des réponses $f(\mathbf{D}_\mathrm{M})$. Pour ce faire, la méthode habituelle est celle du maximum de vraisemblance que nous décrivons ci-dessous. Sous l'hypothèse (A.10), la log-vraisemblance de $(\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f, \sigma_f^2)$ s'écrit

$$l(f(\mathbf{D}_\mathrm{M})|\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f, \sigma_f^2) \propto \; n\log\sigma_f^2 + \log|\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_\mathrm{M})| + \frac{1}{\sigma_f^2}(f(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_f)^\mathrm{T}\Sigma_{\Psi_f}(\mathbf{D}_\mathrm{M})^{-1}\big(f(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\boldsymbol{\beta}_f\big). \quad \text{(A.21)}$$

Pour une valeur de $\boldsymbol{\Psi}_f$ connue, le maximum pour $\boldsymbol{\beta}_f$ est atteint en l'estimateur des moindres carrés pondérés

$$\hat{\boldsymbol{\beta}}_f(\boldsymbol{\Psi}_f) = (h(\mathbf{D}_\mathrm{M})^\mathrm{T}\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_\mathrm{M})^{-1}h(\mathbf{D}_\mathrm{M}))^{-1}h(\mathbf{D}_\mathrm{M})^\mathrm{T}\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_\mathrm{M})^{-1}f(\mathbf{D}_\mathrm{M}). \quad \text{(A.22)}$$

De même, le maximum de σ_f^2 est atteint en

$$\hat{\sigma}_f^2(\boldsymbol{\Psi}_f) = \frac{1}{n}(f(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\hat{\boldsymbol{\beta}}_{\boldsymbol{f}})^\mathrm{T}\Sigma_{\Psi_f}(\mathbf{D}_\mathrm{M})^{-1}(f(\mathbf{D}_\mathrm{M}) - h(\mathbf{D}_\mathrm{M})^\mathrm{T}\hat{\boldsymbol{\beta}}_{\boldsymbol{f}}). \quad \text{(A.23)}$$

En réinjectant (A.22) et (A.23) dans (A.21), on obtient une expression de (A.21) qui ne dépend plus que de $\boldsymbol{\Psi}_f$:

$$l(f(\mathbf{D}_\mathrm{M}|\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f, \sigma_f^2) \propto \; n\log\hat{\sigma}_f^2 + \log|\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_\mathrm{M})|. \quad \text{(A.24)}$$

En injectant

$$\hat{\boldsymbol{\Psi}}_f \in \operatorname*{argmin}_{\boldsymbol{\Psi}_f} n\log\hat{\sigma}_f^2(\boldsymbol{\Psi}_f) + \log|\Sigma_{\boldsymbol{\Psi}_f}(\mathbf{D}_\mathrm{M})|$$

dans (A.22) et (A.23), on obtient les estimateurs du maximum de vraisemblance de $\boldsymbol{\beta}_f$, σ_f^2 et $\boldsymbol{\Psi}_f$ donnés respectivement par $\hat{\boldsymbol{\beta}}_f(\hat{\boldsymbol{\Psi}}_f)$, $\hat{\sigma}_f^2(\hat{\boldsymbol{\Psi}}_f)$ et $\hat{\boldsymbol{\Psi}}_f$. Le prédicteur (A.14), dans lequel $\boldsymbol{\beta}_f$, σ_f^2 et $\boldsymbol{\Psi}_f$ sont remplacés par $\hat{\boldsymbol{\beta}}_f(\hat{\boldsymbol{\Psi}}_f)$, $\hat{\sigma}_f^2(\hat{\boldsymbol{\Psi}}_f)$ et $\hat{\boldsymbol{\Psi}}_f$, est appelé EBLUP (*Empirical Best Linear Unbiaised Predictor*) même s'il n'est ni linéaire, ni sans biais (Santner et al., 2003).

Prise en compte de l'incertitude sur $(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)$ La technique d'estimation décrite précédemment néglige l'incertitude affectant les paramètres $(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)$ en considérant des estimateurs ponctuels de ces quantités, communément appelés estimateurs « plug-in ». Une manière plus adéquate de tenir compte de la méconnaissance de $(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)$ consiste en un traitement bayésien de ces quantités en considérant une loi *a priori* $\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)$. Dans ce cas, les expressions (A.14) et (A.15) sont remplacées par

$$\mathbb{E}[f(\mathbf{v})|f(\mathbf{D}_\mathrm{M})] = \int_{\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f} \mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v})\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f|f(\mathbf{D}_\mathrm{M}))\mathrm{d}\boldsymbol{\beta}_f\mathrm{d}\sigma_f^2\mathrm{d}\boldsymbol{\Psi}_f \quad \text{(A.25)}$$

et

$$Cov[f(\mathbf{v}), f(\mathbf{v}')|f(\mathbf{D}_\mathrm{M})] = \mathbb{E}[V_{\sigma_f^2, \boldsymbol{\Psi}_f}(\mathbf{v}, \mathbf{v}')] + Cov\big(\mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v}), \mu_{\boldsymbol{\beta}_f, \boldsymbol{\Psi}_f}(\mathbf{v}')\big) \quad \text{(A.26)}$$

calculée par rapport à $\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f|f(\mathbf{D}_\mathrm{M}))$. Il a été observé en comparant (A.26) lorsque $\mathbf{v} = \mathbf{v}'$ à (A.15) que l'estimation « plug-in » sous estime l'incertitude de prédiction contrairement à l'approche bayésienne (Handcock et Stein, 1993; Helbert et al., 2009). Lorsque l'on ne dispose pas d'information *a priori*, certaines lois $\pi(\boldsymbol{\beta}_f, \sigma_f^2|\boldsymbol{\Psi}_f)$ bien choisies permettent d'obtenir une forme analytique de la loi *a posteriori* $\pi(\boldsymbol{\beta}_f, \sigma_f^2|\boldsymbol{\Psi}_f, f(\mathbf{D}_\mathrm{M}))$ (voir annexe A.5).

Validation du processus gaussien Pour quantifier les capacités du prédicteur moyen $\mu_{\boldsymbol{\beta}_f,\boldsymbol{\Psi}_f}(\mathbf{v})$ à reproduire f, le coefficient de détermination peut être calculé à partir d'une base de test de taille N. Le coefficient de prédictivité Q^2 s'écrit :

$$Q^2 = 1 - \frac{\sum_{i=1}^{N}\left(f(\mathbf{v}_i) - \mu_{\boldsymbol{\beta}_f,\boldsymbol{\Psi}_f}(\mathbf{v}_i)\right)^2}{\sum_{i=1}^{N}\left(f(\mathbf{v}_i) - \bar{f}\right)^2} \tag{A.27}$$

où $\bar{f} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{v}_i)$. Dans un cas réel, la taille N de la base de test est souvent réduite étant donné le coût élevé des simulations. Par conséquent, la confiance que l'on peut attribuer à la valeur calculée du Q^2 est faible car très dépendante des N configurations choisies. Une approche alternative souvent utilisée en pratique consiste à calculer un Q^2 par validation croisée. Un autre critère couramment utilisé est la racine carrée de l'écart quadratique moyen,

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f(\mathbf{v}_i) - \mu_{\boldsymbol{\beta}_f,\boldsymbol{\Psi}_f}(\mathbf{v}_i)\right)^2}. \tag{A.28}$$

Des techniques de validation plus sophistiquées de l'émulateur processus gaussien sont proposées dans Iooss et al. (2010) et Bastos et O'Hagan (2008).

Processus gaussien « bruité » Il est possible d'approcher f par un processus gaussien à partir de réponses « bruitées » y_i de la forme

$$y_i = f(\mathbf{v}_i) + \epsilon_i \tag{A.29}$$

où

$$\epsilon_i \sim \mathscr{E}_i \underset{i.i.d.}{=} \mathcal{N}(0, \lambda^2) \tag{A.30}$$

sont des lois gaussiennes centrées, indépendantes et identiquement distribuées (i.i.d.), de variance λ^2. Il suffit de remplacer $\Sigma_{\boldsymbol{\Psi}_f}(.)$ par $\Sigma_{\boldsymbol{\Psi}_f}(.) + k\mathrm{I}$ où $k := \lambda^2\sigma^{-2}$. Dans ce cas, la moyenne *a posteriori* (A.14) n'interpole pas les réponses $\{y_i\}_i$.

Remarque Dans le cadre habituel de réponses $f(\mathbf{D}_\mathrm{M})$ déterministes, un bruit de variance λ^2 très faible appelé « effet pépite » est souvent utilisé afin d'obtenir une matrice $\Sigma_{\boldsymbol{\Psi}_f}(.)+\lambda^2\mathrm{I}$ mieux conditionnée que $\Sigma_{\boldsymbol{\Psi}_f}(.)$, notamment lorsque des configurations d'apprentissages $\mathbf{v},\mathbf{v}' \in \mathbf{D}_\mathrm{M}$ sont très proches. Contrairement au cas des réponses bruitées, la propriété d'interpolation est conservée (Roustant et al., 2012).

Pour davantage de détails à propos de l'émulateur processus gaussien et son usage en simulation numérique, nous conseillons les ouvrages de Santner et al. (2003) et Rasmussen et Williams (2006).

A.4 Les plans d'expériences

Les méthodes destinées à la validation des codes de calcul présentées dans ce mémoire sont fondées sur l'émulation par processus gaussien, dont la construction requiert un nombre limité de simulations du code $f(\mathbf{D}_\mathrm{M})$. La problématique qui consiste à choisir en quelles configurations des variables d'entrée $\mathbf{D}_\mathrm{M}$ doivent être exécutées ces simulations est appelée planification d'expériences numériques ; et $\mathbf{D}_\mathrm{M}$ est appelé le plan d'expériences numériques (Koehler et Owen, 1996). La capacité de l'émulateur à approcher

précisément le code sur l'espace de ses variables d'entrée $\mathcal{V}$ dépend fortement du type de plan d'expériences utilisé. Dans cette section, nous donnons un aperçu des enjeux en planification d'expériences numériques d'un code de calcul.

A.4.1 Plans d'expériences remplissant l'espace des entrées

Lorsque l'objectif consiste à prédire au mieux le code de calcul pour tout $\mathbf{v} \in \mathcal{V}$, on cherche couramment à construire un plan d'expériences $\mathbf{D}_{\mathrm{M}}$ disposant de bonnes propriétés de remplissage de $\mathcal{V}$ au sens de critères géométriques et(ou) statistiques (appelé *Space Filling Design*). Dans la suite de cette section, on appelle point de l'espace des entrées $\mathcal{V}$ la valeur donnée à la variable d'entrée $\mathbf{v}$, faisant du plan d'expériences $\mathbf{D}_{\mathrm{M}}$ un ensemble de points de $\mathcal{V}$. Le critère géométrique le plus utilisé consiste à maximiser la distance entre toutes les paires de points. Il est appelé plan *Maximin*. D'autres critères, que l'on cherche cette fois à minimiser, sont basés sur la notion de *discrépance* qui quantifie l'écart entre $\mathbf{D}_{\mathrm{M}}$ et l'uniformité théorique sur $\mathcal{V}$. Récemment les plans d'expériences à *énergie minimale* font apparaître pour la première fois à notre connaissance des liens entre les critères de discrépance et le critère Maximin (Joseph et al., 2015). Il est souvent d'usage de maximiser un des critères présentés précédemment dans la classe des LHD (*Latin Hypercube Design*) qui ont la bonne propriété d'admettre des projections uniformes sur chacune des composantes de $\mathbf{v}$. D'ailleurs, les plans d'expériences à bon remplissage de $\mathcal{V}$ que nous utilisons dans la thèse sont des LHD Maximin, construit à l'aide de la librairie *DiceDesign* (Damblin et al., 2013; Morris et Mitchell, 1995).

Pour une revue détaillée des plans d'expériences à bon remplissage de $\mathcal{V}$, nous conseillons l'article de Pronzato et Muller (2012) ainsi que le mémoire de thèse de Franco (2008) axé sur les plans en grande dimension.

A.4.2 Plans d'expériences orientés vers un objectif spécifique

De plus en plus de travaux récents s'intéressent à établir de nouvelles stratégies de planification d'expériences numériques en fonction de l'objectif poursuivi. Ces stratégies sont souvent adaptatives (séquentielles) car elles s'appuient sur l'émulateur processus gaussien construit à l'étape k pour déterminer les simulations à exécuter à l'étape $k+1$. Par exemple, Sacks et al. (1989) proposent de réduire séquentiellement l'incertitude d'un émulateur processus gaussien en effectuant des simulations dans les zones de $\mathcal{V}$ où l'émulateur admet une grande variance. Jones et al. (1998) proposent un algorithme séquentiel pour trouver l'optimum d'un code de calcul boîte noire coûteux en maximisant le critère EI (*Expected Improvement*). Bect et al. (2012) développent un critère SUR (*Stepwise Uncertainty Reduction*) pour estimer à moindre coût une probabilité de défaillance. Dans sa thèse, Kumar (2008) présente des algorithmes séquentiels pour le calage de codes coûteux. Dans le chapitre 4, nous proposons également de nouveaux plans d'expériences pour cet objectif.

A.5 Inférence bayésienne du processus aléatoire gaussien

Dans cette section, notons pour faciliter la lisibilité des calculs $\mathbf{f}_{\mathrm{M}} := f(\mathbf{D}_{\mathrm{M}})$ et $\mathbf{h}_{\mathrm{M}} := h(\mathbf{D}_{\mathrm{M}})$. Nous allons calculer la loi *a priori* $\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f | f(\mathbf{D}_{\mathrm{M}}))$ pour des classes de loi *a*

priori $\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)$ bien choisies. La formule de Bayes s'écrit :

$$\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f | f(\mathbf{D}_\mathrm{M})) = \frac{\mathscr{L}(f(\mathbf{D}_\mathrm{M})|\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)}{\int_{\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f} \mathscr{L}(f(\mathbf{D}_\mathrm{M})|\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)\pi(\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f)\mathrm{d}\boldsymbol{\beta}_f \mathrm{d}\sigma_f^2 \mathrm{d}\boldsymbol{\Psi}_f} \tag{A.31}$$

avec

$$\mathscr{L}(\mathbf{f}_\mathrm{M}|\boldsymbol{\beta}_f, \sigma_f^2, \boldsymbol{\Psi}_f) = \frac{|\Sigma_{\boldsymbol{\Psi}_f}|^{-1/2}}{(2\pi)^{\mathrm{M}/2}(\sigma_f^2)^{\mathrm{M}/2}} \exp -\frac{1}{2\sigma_f^2}(\mathbf{f}_\mathrm{M} - \mathbf{h}_\mathrm{M}^\mathrm{T}\boldsymbol{\beta}_f)^\mathrm{T} \Sigma_{\boldsymbol{\Psi}_f}^{-1}(\mathbf{f}_\mathrm{M} - \mathbf{h}_\mathrm{M}^\mathrm{T}\boldsymbol{\beta}_f). \tag{A.32}$$

Nous supposons désormais que le vecteur de paramètres $\boldsymbol{\Psi}_f$ spécifiant la covariance est connu et nous utilisons la notation $\Sigma := \Sigma_{\boldsymbol{\Psi}_f}$.

Loi a priori gamma-normale Pour pouvoir obtenir certaines expressions analytiques des lois *a posteriori*, on peut considérer la distribution *a priori* hiérarchique suivante :

$$\pi(\boldsymbol{\beta}_f, \sigma_f^2) = \pi(\boldsymbol{\beta}_f|\sigma_f^2)\pi(\sigma_f^2) \tag{A.33}$$

avec

$$\boldsymbol{\beta}_f|\sigma^2 \sim \mathscr{N}(\mathbf{b}_0, \sigma_f^2 \mathrm{V}_0) \tag{A.34}$$

et

$$\sigma_f^2 \sim \mathscr{IG}(a, b). \tag{A.35}$$

Le paramètre a désigne le paramètre de forme de la loi inverse-gamma et b désigne le paramètre d'intensité, de telle sorte que la densité de probabilité de σ_f^2 s'écrit

$$\pi(\sigma_f^2) = \frac{b^a}{\Gamma(a)}\left(\frac{1}{\sigma_f^2}\right)^{a+1} \exp -\frac{b}{\sigma_f^2} \tag{A.36}$$

Nous allons calculer la loi *a posteriori* $\pi(\boldsymbol{\beta}_f, \sigma_f^2|\mathbf{f}_\mathrm{M})$ en la décomposant sous la forme hiérarchique

$$\pi(\boldsymbol{\beta}_f, \sigma_f^2|\mathbf{f}_\mathrm{M}) = \pi(\boldsymbol{\beta}_f|\sigma_f^2, \mathbf{f}_\mathrm{M})\pi(\sigma_f^2|\mathbf{f}_\mathrm{M}) \tag{A.37}$$

avec d'une part,

$$\pi(\boldsymbol{\beta}_f|\sigma_f^2, \mathbf{f}_\mathrm{M}) \propto \mathscr{L}(\mathbf{f}_\mathrm{M}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2) \tag{A.38}$$

et d'autre part,

$$\begin{aligned}\pi(\sigma_f^2|\mathbf{f}_\mathrm{M}) = \frac{\pi(\sigma_f^2, \mathbf{f}_\mathrm{M})}{\pi(\mathbf{f}_\mathrm{M})} \propto \pi(\sigma_f^2, \mathbf{f}_\mathrm{M}) &= \frac{\pi(\boldsymbol{\beta}_f, \sigma_f^2, \mathbf{f}_\mathrm{M})}{\pi(\boldsymbol{\beta}_f|\sigma_f^2, \mathbf{f}_\mathrm{M})} \\ &= \frac{\mathscr{L}(\mathbf{f}_\mathrm{M}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\pi(\sigma_f^2)}{\pi(\boldsymbol{\beta}_f|\sigma_f^2, \mathbf{f}_\mathrm{M})}.\end{aligned} \tag{A.39}$$

Commençons par calculer (A.38) :

$$\begin{aligned}\mathscr{L}(\mathbf{f}_\mathrm{M}|\boldsymbol{\beta}_f, \sigma_f^2)\pi^0(\boldsymbol{\beta}_f|\sigma_f^2) = &\frac{1}{(2\pi)^{\mathrm{M}/2}(\sigma_f^2)^{\mathrm{M}/2}|\Sigma|^{1/2}} \\ &\exp -\frac{1}{2\sigma_f^2}(\mathbf{f}_\mathrm{M} - \mathbf{h}_\mathrm{M}^\mathrm{T}\boldsymbol{\beta}_f)^\mathrm{T}\Sigma^{-1}(\mathbf{f}_\mathrm{M} - \mathbf{h}_\mathrm{M}^\mathrm{T}\boldsymbol{\beta}_f) \times \\ &\frac{1}{(2\pi)^{p/2}(\sigma_f^2)^{p/2}|\mathrm{V}_0|^{1/2}} \exp -\frac{1}{2\sigma_f^2}(\boldsymbol{\beta}_f - \mathbf{b}_0)^\mathrm{T}\mathrm{V}_0^{-1}(\boldsymbol{\beta}_f - \mathbf{b}_0).\end{aligned}$$

Si l'on développe, on obtient :

$$\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi^0(\boldsymbol{\beta}_f|\sigma_f^2) = \frac{1}{(2\pi)^{\frac{\mathrm{M}+p}{2}}(\sigma_f^2)^{\frac{\mathrm{M}+p}{2}}|\Sigma|^{1/2}|\mathrm{V}_0|^{1/2}}\exp-\frac{b_0^{\mathrm{T}}\mathrm{V}_0^{-1}b_0}{2\sigma_f^2}$$
$$\exp-\frac{1}{2\sigma_f^2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}\times\exp-\frac{1}{2}\boldsymbol{\beta}_f^{\mathrm{T}}\tilde{\mathrm{A}}^{-1}\boldsymbol{\beta}_f\times\exp\tilde{\mu}^{\mathrm{T}}\boldsymbol{\beta}_f$$
$$\exp-\frac{\boldsymbol{\beta}_f^{\mathrm{T}}\mathrm{V}_0^{-1}\boldsymbol{\beta}_f}{2\sigma_f^2}\exp\frac{b_0^{\mathrm{T}}\mathrm{V}_0^{-1}\boldsymbol{\beta}_f}{\sigma_f^2}$$

avec $\tilde{\mathrm{A}}^{-1}=\frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}}{\sigma_f^2}$ et $\tilde{\mu}=\frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}}{\sigma_f^2}$. Et donc,

$$\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi^0(\boldsymbol{\beta}_f|\sigma_f^2) = \frac{1}{(2\pi)^{\frac{\mathrm{M}+p}{2}}(\sigma_f^2)^{\frac{\mathrm{M}+p}{2}}|\Sigma|^{1/2}|\mathrm{V}_0|^{1/2}}\exp-\frac{b_0^{\mathrm{T}}\mathrm{V}_0^{-1}\mathbf{b}_0}{2\sigma_f^2}$$
$$\exp-\frac{1}{2\sigma_f^2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}\times\exp-\frac{1}{2}\boldsymbol{\beta}_f^{\mathrm{T}}\mathrm{A}^{-1}\boldsymbol{\beta}_f\times\exp\mu^{\mathrm{T}}\boldsymbol{\beta}_f$$

avec $\mathrm{A}^{-1}=\frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}}{\sigma_f^2}+\frac{\mathrm{V}_0^{-1}}{\sigma_f^2}$ et $\mu=\frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}}{\sigma_f^2}+\frac{\mathrm{V}_0^{-1}}{\sigma_f^2}\mathbf{b}_0$. En utilisant le lemme ci-dessous on en déduit que

$$\boldsymbol{\beta}_f|\mathbf{f}_{\mathrm{M}},\sigma_f^2\sim\mathscr{N}(\mathrm{A}\mu,\mathrm{A}). \tag{A.40}$$

Lemme A.5.0.1 *Pour tout* $\mathbf{x}\in\mathbb{R}^n$ *et pour tout* $\mathrm{A}\in\mathrm{M}_{n,n}(\mathbb{R})$ *matrice symétrique définie positive, si*

$$f(\mathbf{y})\propto\exp\big(-\frac{1}{2}\mathbf{y}^{\mathrm{T}}\mathrm{A}^{-1}\mathbf{y}+\mathbf{x}^{\mathrm{T}}\mathbf{y}\big)$$

alors $\mathbf{Y}\sim\mathscr{N}(\mathrm{A}\mathbf{x},\mathrm{A})$.

De plus, on peut calculer la constante de normalisation

$$\int\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\mathrm{d}\boldsymbol{\beta}_f=\frac{\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)}{\pi(\boldsymbol{\beta}_f|\sigma_f^2,\mathbf{f}_{\mathrm{M}})}. \tag{A.41}$$

Il vient,

$$\int\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\mathrm{d}\boldsymbol{\beta}_f=\frac{(\sqrt{2\pi})^p(\sigma_f^2)^{p/2}|(\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}+\mathrm{V}_0^{-1})^{-1}|^{1/2}}{(2\pi)^{\frac{\mathrm{M}+p}{2}}(\sigma_f^2)^{\frac{\mathrm{M}+p}{2}}|\Sigma|^{1/2}|\mathrm{V}_0|^{1/2}}$$
$$\exp-\frac{\mathbf{b}_0^{\mathrm{T}}\mathrm{V}_0^{-1}\mathbf{b}_0}{2\sigma_f^2}\exp-\frac{1}{2\sigma_f^2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}\times\exp\frac{1}{2}\mu^{\mathrm{T}}\mathrm{A}\mu$$

puis,

$$\int\mathscr{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\mathrm{d}\boldsymbol{\beta}_f=\frac{|(\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}+\mathrm{V}_0^{-1})^{-1}|^{1/2}}{(2\pi)^{\frac{n}{2}}(\sigma_f^2)^{\frac{n}{2}}|\Sigma|^{1/2}|\mathrm{V}_0|^{1/2}}\exp-\frac{\mathbf{b}_0^{\mathrm{T}}\mathrm{V}_0^{-1}\mathbf{b}_0}{2\sigma_f^2}$$
$$\exp-\frac{1}{2\sigma_f^2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}\times\exp\frac{1}{2}\mu^{\mathrm{T}}\mathrm{A}\mu. \tag{A.42}$$

En combinant (A.39) et (A.41) et en utilisant l'expression (A.42), $\pi(\sigma_f^2|\mathbf{f}_M)$ s'écrit à une constante de normalisation près :

$$\pi(\sigma_f^2|\mathbf{f}_M) \propto \left(\frac{1}{\sigma_f^2}\right)^{n/2+a+1} \exp -\frac{\mathbf{b}_0^T V_0^{-1}\mathbf{b}_0}{2\sigma_f^2} \exp -\frac{b}{\sigma_f^2} \exp -\frac{1}{2\sigma_f^2}\mathbf{f}_M^T \Sigma^{-1}\mathbf{f}_M \times \exp \frac{1}{2}\mu^T A\mu. \quad \text{(A.43)}$$

Commençons par développer $\exp \frac{1}{2}\mu^T A\mu$:

$$\exp \frac{1}{2}\mu^T A\mu = \exp\left(\frac{1}{2\sigma^2}\left[\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M + V_0^{-1}\mathbf{b}_0\right]^T \left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1}\left[\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M + V_0^{-1}\mathbf{b}_0\right]\right).$$

En utilisant l'identité $(A+C)^{-1} = A^{-1} + A^{-1}(A^{-1}+C^{-1})^{-1}A^{-1}$, il vient

$$\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1} = \left[V_0 - V_0[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}V_0\right]. \quad \text{(A.44)}$$

Par suite,

$$\mathbf{b}_0^T V_0^{-1}\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1} V_0^{-1}\mathbf{b}_0 = \mathbf{b}_0^T V_0^{-1}\mathbf{b}_0 - \mathbf{b}_0^T[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}\mathbf{b}_0.$$

Soit $\hat{\boldsymbol{\beta}} := (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M$, alors

$$\mathbf{f}_M^T\Sigma^{-1}\mathbf{h}_M^T\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1} V_0^{-1}\mathbf{b}_0 = \hat{\boldsymbol{\beta}}^T[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}\mathbf{b}_0$$

et

$$\mathbf{b}_0^T V_0^{-1}\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M = \mathbf{b}_0^T[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}\hat{\boldsymbol{\beta}}.$$

Enfin, en remarquant que

$$\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1} = \left[(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1} - (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + \mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T V_0 \mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\right]$$

on peut aboutir en développant les calculs à

$$\mathbf{f}_M^T\Sigma^{-1}\mathbf{h}_M^T\left[\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T + V_0^{-1}\right]^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M = \mathbf{f}_M^T\Sigma^{-1}\mathbf{h}_M^T(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M - \hat{\boldsymbol{\beta}}^T[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}\hat{\boldsymbol{\beta}}.$$

Par suite,

$$\exp \frac{1}{2}\mu^T A\mu \times \exp -\frac{\mathbf{b}_0^T V_0^{-1}\mathbf{b}_0}{2\sigma_f^2} = \exp -\frac{1}{2\sigma_f^2}\Big((\mathbf{b}_0 - \hat{\boldsymbol{\beta}})^T[V_0 + (\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}]^{-1}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) - \mathbf{f}_M^T\Sigma^{-1}\mathbf{h}_M^T(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M\Big).$$

Finalement, en repartant de l'équation (A.43), on obtient

$$\pi(\sigma_f^2|\mathbf{f}_M) \propto \left(\frac{1}{\sigma_f^2}\right)^{M/2+a-1} \exp -\frac{Q}{\sigma_f^2} \quad \text{(A.45)}$$

avec

$$\mathrm{Q} = b + \frac{1}{2}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}})^{\mathrm{T}}[\mathrm{V}_0 + (\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}})^{-1}]^{-1}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) - \frac{1}{2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}(\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}})^{-1}\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}} + \frac{1}{2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}, \quad \text{(A.46)}$$

qui peut se réécrire

$$\mathrm{Q} = b + \frac{1}{2}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}})^{\mathrm{T}}[\mathrm{V}_0 + (\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}})^{-1}]^{-1}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) + (\mathbf{f}_{\mathrm{M}} - \mathbf{h}_{\mathrm{M}}^{\mathrm{T}}\hat{\boldsymbol{\beta}})^{\mathrm{T}}\Sigma^{-1}(\mathbf{f}_{\mathrm{M}} - \mathbf{h}_{\mathrm{M}}^{\mathrm{T}}\hat{\boldsymbol{\beta}}). \quad \text{(A.47)}$$

Donc

$$\sigma_f^2|\mathbf{f}_{\mathrm{M}} \sim \mathcal{IG}(\frac{\mathrm{M}}{2} + a, \mathrm{Q}). \quad \text{(A.48)}$$

Enfin, la loi marginale des observations $\pi(\mathbf{f}_{\mathrm{M}})$ peut être calculée. Elle s'écrit

$$\pi(\mathbf{f}_{\mathrm{M}}) = \int_{\sigma_f^2}\int_{\boldsymbol{\beta}_f} \mathcal{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\pi(\sigma_f^2)\mathrm{d}\boldsymbol{\beta}_f\mathrm{d}\sigma_f^2.$$

En utilisant (A.42), on calcule

$$\pi(\mathbf{f}_{\mathrm{M}}) = \frac{b^a \times \Gamma(\frac{\mathrm{M}}{2} + a) \times |(\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}} + \mathrm{V}_0^{-1})^{-1}|^{1/2}}{\Gamma(a) \times \mathrm{Q}^{\frac{\mathrm{M}}{2}+a} \times (2\pi)^{\frac{\mathrm{M}}{2}}|\Sigma|^{1/2}|\mathrm{V}_0|^{1/2}}. \quad \text{(A.49)}$$

Loi a priori objective Considérons maintenant la distribution hiérarchique suivante :

$$\pi(\boldsymbol{\beta}_f, \sigma_f^2) = \pi(\boldsymbol{\beta}_f|\sigma_f^2)\pi(\sigma_f^2) \quad \text{(A.50)}$$

avec

$$\pi(\boldsymbol{\beta}_f)|\sigma^2 \propto 1 \quad \text{(A.51)}$$

et

$$\pi(\sigma_f^2) \propto \frac{1}{(\sigma_f^2)^a}. \quad \text{(A.52)}$$

et pour $a \in \mathbb{R}$. Commençons par calculer (A.38) :

$$\begin{aligned}\mathcal{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2) &= \frac{1}{(2\pi)^{\mathrm{M}/2}(\sigma_f^2)^{\mathrm{M}/2}|\Sigma|^{1/2}}\exp -\frac{1}{2\sigma_f^2}(\mathbf{f}_{\mathrm{M}} - \mathbf{h}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{\beta}_f)^{\mathrm{T}}\Sigma^{-1}(\mathbf{f}_{\mathrm{M}} - \mathbf{h}_{\mathrm{M}}^{\mathrm{T}}\boldsymbol{\beta}_f) \\ &= \frac{1}{(2\pi)^{\mathrm{M}/2}(\sigma_f^2)^{\mathrm{M}/2}\sqrt{|\Sigma|}} \times \exp -\frac{1}{2\sigma_f^2}\mathbf{f}_{\mathrm{M}}^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}} \times \exp -\frac{1}{2}\boldsymbol{\beta}_f^{\mathrm{T}}\mathrm{A}^{-1}\boldsymbol{\beta}_f \times \exp \mu^{\mathrm{T}}\boldsymbol{\beta}_f\end{aligned}$$

avec $\mathrm{A}^{-1} = \frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{h}_{\mathrm{M}}^{\mathrm{T}}}{\sigma_f^2}$ et $\mu = \frac{\mathbf{h}_{\mathrm{M}}\Sigma^{-1}\mathbf{f}_{\mathrm{M}}}{\sigma_f^2}$. En utilisant le lemme, on en déduit que

$$\boldsymbol{\beta}_f|\mathbf{f}_{\mathrm{M}}, \sigma_f^2 \sim \mathcal{N}(\mathrm{A}\mu, \mathrm{A}). \quad \text{(A.53)}$$

De plus, on peut calculer sa constante de normalisation

$$\int \mathcal{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\mathrm{d}\boldsymbol{\beta}_f = \frac{\mathcal{L}(\mathbf{f}_{\mathrm{M}}|\boldsymbol{\beta}_f, \sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)}{\pi(\boldsymbol{\beta}_f|\sigma_f^2, \mathbf{f}_{\mathrm{M}})}. \quad \text{(A.54)}$$

Il vient,

$$\int \mathscr{L}(\mathbf{f}_M|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)d\boldsymbol{\beta}_f = \frac{(2\pi)^{p/2}(\sigma_f^2)^{p/2}|(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}|^{1/2}}{(2\pi)^{M/2}(\sigma_f^2)^{M/2}|\Sigma|^{1/2}}\times \exp-\frac{1}{2\sigma_f^2}\mathbf{f}_M^T\Sigma^{-1}\mathbf{f}_M \times \exp \mu^T A\mu. \quad (A.55)$$

En utilisant l'expression (A.55), $\pi(\sigma_f^2|\mathbf{f}_M)$ s'écrit à une constante de normalisation près :

$$\pi(\sigma_f^2|\mathbf{f}_M) \propto (\sigma_f^2)^{\frac{p-M}{2}-a}\exp-\frac{1}{2\sigma_f^2}\mathbf{f}_M^T\Sigma^{-1}\mathbf{f}_M \exp\frac{1}{2}\mu^T A\mu \quad (A.56)$$

En développant,

$$\begin{aligned}\exp-\frac{1}{2\sigma_f^2}z^T\Sigma^{-1}\mathbf{f}_M\exp\frac{1}{2}\mu^T A\mu &= \exp-\frac{1}{2\sigma_f^2}\mathbf{f}_M^T\Sigma^{-1}\mathbf{f}_M\exp\frac{1}{2\sigma_f^2}\mathbf{f}_M^T\Sigma^{-1}\mathbf{h}_M^T(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1}\mathbf{f}_M\\ &= \exp-\frac{1}{2\sigma_f^2}\mathbf{f}_M^T(\Sigma^{-1}-\Sigma^{-1}\mathbf{h}_M^T(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1})\mathbf{f}_M.\end{aligned}$$

Donc,

$$\sigma_f^2|\mathbf{f}_M \sim \mathscr{IG}(\frac{M-p}{2}-1+a, Q) \quad (A.57)$$

avec

$$\begin{aligned}Q &= \frac{1}{2}\mathbf{f}_M^T(\Sigma^{-1}-\Sigma^{-1}\mathbf{h}_M^T(\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T)^{-1}\mathbf{h}_M\Sigma^{-1})\mathbf{f}_M\\ &= (\mathbf{f}_M-\mathbf{h}_M^T\hat{\boldsymbol{\beta}})^T\Sigma^{-1}(\mathbf{f}_M-\mathbf{h}_M^T\hat{\boldsymbol{\beta}}).\end{aligned}$$

Enfin, la loi marginale des observations $\pi(\mathbf{f}_M)$ peut être calculée. Elle s'écrit

$$\pi(\mathbf{f}_M) = \int_{\sigma_f^2}\int_{\boldsymbol{\beta}_f}\mathscr{L}(\mathbf{f}_M|\boldsymbol{\beta}_f,\sigma_f^2)\pi(\boldsymbol{\beta}_f|\sigma_f^2)\pi(\sigma_f^2)d\boldsymbol{\beta}_f d\sigma_f^2.$$

En utilisant (A.42), on calcule

$$\pi(\mathbf{f}_M) \propto \frac{(2\pi)^{\frac{p-M}{2}}}{(|\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T||\Sigma|)^{1/2}}\times\frac{\Gamma(\frac{M-p}{2}-1+a)}{Q^{(\frac{M-p}{2}-1+a)}}. \quad (A.58)$$

Lorsque $a=1$, on retrouve la loi *a priori* de Jeffreys sur le paramètre d'échelle σ_f^2. Dans ce cas,

$$\pi(\mathbf{f}_M) \propto \frac{(2\pi)^{\frac{p-M}{2}}}{(|\mathbf{h}_M\Sigma^{-1}\mathbf{h}_M^T||\Sigma|)^{1/2}}\times\frac{\Gamma(\frac{M-p}{2})}{Q^{(\frac{M-p}{2})}}. \quad (A.59)$$

Remarque A.5.0.1 *Des expressions analytiques peuvent également être obtenues en considérant les lois a priori suivante (Santner et al., 2003) :*

$$\pi(\boldsymbol{\beta}_f|\sigma_f^2) \propto 1 \text{ et } \sigma_f^2 \sim \mathscr{IG}(a,b) \quad (A.60)$$

mais aussi

$$\boldsymbol{\beta}_f|\sigma^2 \sim \mathscr{N}(\mathbf{b}_0,\sigma_f^2 V_0) \text{ et } \pi(\sigma_f^2) \propto \frac{1}{\sigma_f^2}. \quad (A.61)$$

Si l'on utilise la loi *a priori* (A.61), on obtient facilement comme cas particulier de (A.45) :

$$\pi(\sigma_f^2|\mathbf{f}_M) \propto \left(\frac{1}{\sigma_f^2}\right)^{M/2-1} \exp -\frac{Q}{\sigma_f^2}$$

avec

$$Q = \frac{1}{2}(\mathbf{b}_0 - \hat{\boldsymbol{\beta}})^T [V_0 + (\mathbf{h}_M \Sigma^{-1} \mathbf{h}_M^T)^{-1}]^{-1} (\mathbf{b}_0 - \hat{\boldsymbol{\beta}}) - \frac{1}{2}\mathbf{f}_M^T \Sigma^{-1} \mathbf{h}_M^T (\mathbf{h}_M \Sigma^{-1} \mathbf{h}_M^T)^{-1} \mathbf{h}_M \Sigma^{-1} \mathbf{f}_M + \frac{1}{2}\mathbf{f}_M^T \Sigma^{-1} \mathbf{f}_M. \quad (A.62)$$

Par suite,

$$\sigma_f^2|\mathbf{f}_M \sim \mathcal{IG}(\frac{M}{2}, Q) \quad (A.63)$$

et

$$\pi(\mathbf{f}_M) = \frac{\Gamma(\frac{M}{2}) \times |(\mathbf{h}_M \Sigma^{-1} \mathbf{h}_M^T + V_0^{-1})^{-1}|^{1/2}}{Q^{\frac{M}{2}} \times (2\pi)^{\frac{M}{2}} |\Sigma|^{1/2} |V_0|^{1/2}}. \quad (A.64)$$

Calcul du prédicteur Si l'on suppose le paramètre $\boldsymbol{\Psi}_f$ connu pour chacune des lois *a priori* $\pi(\boldsymbol{\beta}_f, \sigma_f^2)$ étudiées précédemment, la loi *a posteriori* $f(\mathbf{v})|f(\mathbf{D}_M)$ est une loi de Student décentrée obtenue en calculant (A.25) et (A.26) avec $\mathbf{v} = \mathbf{v}'$ (Santner et al., 2003).

Bibliographie

Abrahamsen, P. (1997). A review of gaussian random fields and correlation functions. Technical report, Technical Report 917, Norwegian Computing Center, Box 114, Blindern, N-0314, Oslo, Norway. 124

Bastos, L. et O'Hagan, A. (2008). Diagnostics for Gaussian process emulators. *Technometrics*, 51(4) :425–438. 129

Bect, J., Ginsbourger, D., Li, L., Picheny, V., et Vasquez, E. (2012). Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3) :773–793. 130

Bettinger, R. (2009). *Inversion d'un systÃ¨me par krigeage*. PhD thesis, UniversitÃ© de Nice-Sophia Antipolis. 124

Cressie, N. (1993). *Statistics for Spatial Data, Revised Edition*. Wiley, New York. 124

Currin, C., Mitchell, T., Morris, M., et Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the Statistical Association*, 86(416) :953–963. 126

Damblin, G., Couplet, M., et Iooss, B. (2013). Numerical studies of space filling designs : optimization of latin hypercube samples and subprojection properties. *Journal of simulation*, 7 :276–289. 130

Franco, J. (2008). *Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne. 130

Handcock, M. et Stein, M. (1993). A bayesian analysis of kriging. *Technometrics*, 35(4) :403–410. 128

Helbert, C., Dupuy, D., et Carraro, L. (2009). Assessment of uncertainty in computer experiments, from universal to bayesian kriging. *Applied Stochastic Models in Business and Industry*, 25(2) :99–113. 128

Iooss, B., Boussouf, L., Feuillard, V., et Marrel, A. (2010). Numerical studies of the metamodel fitting and validation processes. *International Journal on Advances in Systems and Measurements*, 3 :11–21. 129

Jones, D., Schonlau, M., et Welch, W. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4) :455–492. 130

Joseph, V., Dasgupta, T., Tuo, R., et Wu, C. J. (2015). Sequential exploration of complex surfaces using minimum energy designs. *Technometrics*, 57(1) :64–74. 130

Koehler, J. et Owen, A. (1996). Computer experiments. *Handbook of Statistics, Vol.13*. 129

Kumar, A. (2008). *Sequential calibration of computer models*. PhD thesis, The Ohio State University. 130

Morris, D. et Mitchell, J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43 :381–402. 130

Pronzato, L. et Muller, W. (2012). Design of computer experiments : space filling and beyond. *Statistics and Computing*, 22(3) :681–701. 130

Rasmussen, C. et Williams, C. (2006). *Gaussian Processes for Machine Learning*. the MIT press. 129

Roustant, R., Ginsbourger, D., et Deville, Y. (2012). Dicekriging, Diceoptim : Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. *Journal of Statistical Software*, 51(1) :1–55. 129

Sacks, J., W.J., W., Mitchell, T., et Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4) :409–423. 130

Santner, T., Williams, B., et Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag. 124, 128, 129, 135, 136

Annexe B

L'analyse bayésienne linéaire

Dans cette annexe au mémoire, nous donnons quelques éléments à propos de la théorie bayésienne linéaire (Goldstein et Wooff, 2007). Elle peut être utilisée pour valider un code de calcul $y_{\boldsymbol{\theta}}(\mathbf{x})$ pour lequel les méthodes bayésiennes classiques étudiées dans ce mémoire ne sont pas applicables. C'est le cas lorsque $\boldsymbol{\theta}$ est un paramètre de grande dimension à estimer et/ou lorsque les variables d'entrée et/ou les réponses du code appartiennent à des espaces de grande dimension.

Notons $\mathbf{B} = (\mathrm{B}_1, \cdots, \mathrm{B}_r)$ et $\mathbf{D} = (\mathrm{D}_1, \cdots, \mathrm{D}_s)$ deux collections de variables aléatoires où $\mathbf{B}$ désigne le vecteur des inconnus et $\mathbf{D}$ désigne le vecteur des observables. Contrairement à une analyse bayésienne classique qui consiste à mettre à jour à partir de $\mathbf{D}$ l'information *a priori* sur $\mathbf{B}$ encodée sous la forme d'une distribution de probabilité, la théorie de l'analyse bayésienne linéaire s'appuie seulement sur les spécifications des moyennes, variances et covariances de ces quantités aléatoires. En effet, il peut être beaucoup trop ambitieux de vouloir exprimer les distributions de probabilité de $\mathbf{B}$ et $\mathbf{D}$. Plus raisonnablement, les experts sont souvent plus à même d'afficher des croyances sur les deux premiers moments de $\mathbf{B}$.

Notons $\mathbb{E}[\mathbf{B}]$, $\mathbb{E}[\mathbf{D}]$ les vecteurs d'espérance de $\mathbf{B}$ et $\mathbf{D}$, $\mathbb{V}[\mathbf{B}]$ et $\mathbb{V}[\mathbf{D}]$ les matrices de variance-covariance de ces deux vecteurs et $\mathrm{C}ov(\mathbf{B},\mathbf{D})$ la matrice de covariance croisée. Un moyen d'améliorer nos croyances sur $\mathbf{B}$ en utilisant cette structure de croyances *a priori* consiste à considérer la combinaison linéaire de $\mathbf{D}$, $\sum_{i=0}^{s} a_i \mathrm{D}_i$ qui minimise

$$\mathbb{E}\left(\left[\mathrm{B}_i - \sum_{j=0}^{s} a_j \mathrm{D}_j\right]^2\right) \quad \text{pour} \quad i = 1, \cdots, r. \tag{B.1}$$

où $\mathrm{D}_0 = 1$.

Definition B.0.0.1 *L'estimateur linéaire bayésien en réponse au problème de minimisation précédent est appelé espérance ajustée et vaut*

$$\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i] = \mathbb{E}[\mathrm{B}_i] + \mathrm{C}ov(\mathrm{B}_i, \mathbf{D})\mathbb{V}[\mathbf{D}]^{-1}(\mathbf{D} - \mathbb{E}[\mathbf{D}]). \tag{B.2}$$

Notons que lorsque D n'est pas inversible, elle est remplacée par la matrice pseudo-inverse de Moore-Penrose. En calculant (B.2) pour les r composantes de $\mathbf{B}$, on peut définir naturellement l'espérance ajustée du vecteur $\mathbf{B}$:

$$\mathbb{E}_{\mathbf{D}}[\mathbf{B}] = \mathbb{E}[\mathbf{B}] + \mathrm{C}ov(\mathbf{B},\mathbf{D})\mathbb{V}[\mathbf{D}]^{-1}(\mathbf{D} - \mathbb{E}[\mathbf{D}]). \tag{B.3}$$

Tant que $\mathbf{D}$ n'est pas observé, l'estimateur (B.3) est une variable aléatoire dont la variance peut-être interprétée en décomposant la variance de $\mathbf{B}$ en la somme du vecteur résiduel

$\mathbf{B} - \mathbb{E}_{\mathbf{D}}[\mathbf{B}]$ et de $\mathbb{E}_{\mathbf{D}}[\mathbf{B}]$. Comme $\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]$ est la projection orthogonale de B_i sur $< \mathbf{D} >$, ces deux vecteurs sont décorrélés, d'où

$$\mathbb{V}[\mathrm{B}_i] = \mathbb{V}[\mathrm{B}_i - \mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]] + \mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]] \tag{B.4}$$

avec

$$\mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]] = \mathrm{C}ov(\mathrm{B}_i, \mathbf{D})\mathbb{V}[\mathbf{D}]^{-1}\mathrm{C}ov(\mathbf{D}, \mathrm{B}_i) \tag{B.5}$$

et

$$\begin{aligned}\mathbb{V}[\mathrm{B}_i - \mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]] =& \mathbb{E}[(\mathrm{B}_i - \mathbb{E}_{\mathbf{D}}[\mathrm{B}_i])^2] \\ =& \mathbb{V}[\mathrm{B}_i] - \mathrm{C}ov(\mathbf{D}, \mathrm{B}_i)\mathbb{V}[\mathbf{D}]^{-1}\mathrm{C}ov(\mathbf{D}, \mathrm{B}_i).\end{aligned}$$

Dans Goldstein et Wooff (2007), la quantité $\mathbb{V}[\mathrm{B}_i - \mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]]$ est notée $\mathbb{V}_{\mathbf{D}}[\mathrm{B}_i]$. Elle est appelée *variance résiduelle* (*adjusted variance*) et $\mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]]$ est appelée *variance expliquée* (*resolved variance*). Le ratio

$$\mathrm{R}_{\mathbf{D}}(\mathrm{B}_i) = \frac{\mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathrm{B}_i]]}{\mathbb{V}[\mathrm{B}_i]} \tag{B.6}$$

est appelé résolution. Il représente le pourcentage de variance de B_i expliquée par l'observation de $\mathbf{D}$. Si la résolution est proche de 0, cela signifie que le vecteur $\mathbf{D}$ est peu informatif sur B_i relativement aux croyances *a priori* que nous avons formulées. Les équations (B.4) et (B.6) peuvent être naturellement étendues au vecteur $\mathbf{B}$. On obtient alors les équations matricielles suivantes :

$$\begin{aligned}\mathbb{V}[\mathbf{B}] =& \mathbb{V}[\mathbf{B} - \mathbb{E}_{\mathbf{D}}[\mathbf{B}]] + \mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathbf{B}]] \\ =& \mathbb{V}_{\mathbf{D}}[\mathbf{B}] + \mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathbf{B}]]\end{aligned}$$

avec

$$\mathbb{V}_{\mathbf{D}}[\mathbf{B}] = \mathbb{V}[\mathbf{B}] - \mathrm{C}ov(\mathbf{D}, \mathbf{B})\mathbb{V}[\mathbf{D}]^{-1}\mathrm{C}ov(\mathbf{D}, \mathbf{B}) \tag{B.7}$$

et

$$\mathbb{V}[\mathbb{E}_{\mathbf{D}}[\mathbf{B}]] = \mathrm{C}ov(\mathbf{B}, \mathbf{D})\mathbb{V}[\mathbf{D}]^{-1}\mathrm{C}ov(\mathbf{D}, \mathbf{B}). \tag{B.8}$$

Pour savoir si une réduction significative de nos croyances *a priori* sur $\mathbf{B}$ peut être obtenue en observant $\mathbf{D}$, *l'analyse canonique des corrélations* est très utile. Elle nous permet de comprendre et d'interpréter les résultats d'une analyse bayésienne linéaire qui peuvent parfois surprendre au premier abord, comme l'illustre le chapitre 1 de Goldstein et Wooff (2007). Dans cette ouvrage, des mesures de diagnostic ont aussi été définies afin de détecter un conflit éventuel entre les données et les croyances *a priori*.

Bibliographie

Goldstein, D. et Wooff, D. (2007). *Bayes Linear Statistics, Theory and Methods*. Wiley. 139, 140

Contributions statistiques au calage et à la validation des codes de calcul

Résumé La validation des codes de calcul a pour but d'évaluer l'incertitude de prédiction d'un système physique à partir d'un code de calcul l'approchant et des mesures physiques disponibles. D'une part, le code peut ne pas être une représentation exacte de la réalité. D'autre part, le code peut être entaché d'une incertitude affectant la valeur de certains de ses paramètres, dont l'estimation est appelée « calage de code ». Après avoir dressé un état de l'art unifié des principales procédures de calage et de validation des codes de calcul, nous proposons plusieurs contributions à ces deux problématiques lorsque le code est appréhendé comme une fonction boîte noire coûteuse. D'abord, nous développons une technique bayésienne de sélection de modèle pour tester l'existence d'une fonction d'erreur entre les réponses du code et le système physique, appelée « erreur de code ». Ensuite, nous présentons de nouveaux algorithmes destinés à la construction de plans d'expériences séquentiels afin de rendre plus précis le calage d'un code de calcul basé sur l'émulation par un processus gaussien. Enfin, nous validons un code de calcul utilisé pour prédire la consommation énergétique d'un bâtiment au cours d'une période de temps. Nous utilisons les résultats de l'étude de validation pour apporter une solution à un problème de statistique décisionnelle dans lequel un fournisseur d'électricité doit s'engager auprès de ses clients sur des prévisions moyennes de consommation. En utilisant la théorie bayésienne de la décision, des estimateurs ponctuels optimaux sont calculés.

Mots-clés Validation d'un code de calcul, Calage bayésien, Sélection bayésienne de modèle, Émulation par processus gaussien, Théorie de la décision bayésienne.

Statistical contributions to code calibration and validation

Abstract Code validation aims at assessing the uncertainty affecting the predictions of a physical system by using both the outputs of a computer code which attempt to reproduce it and the available field measurements. In the one hand, the code may be not a perfect representation of the reality. In the other hand, some code parameters can be uncertain and need to be estimated: this issue is referred to as code calibration. After having provided a unified view of the main procedures of code validation, we propose several contributions for solving some issues arising in computer codes which are both costly and considered as black-box functions. First, we develop a Bayesian testing procedure to detect whether or not a discrepancy function, called code discrepancy, has to be taken into account between the code outputs and the physical system. Second, we present new algorithms for building sequential designs of experiments in order to reduce the error occurring in the calibration process based on a Gaussian process emulator. Lastly, a validation procedure of a thermal code is conducted as the preliminary step of a decision problem where an energy supplier has to commit for an overall energy consumption forecast to customers. Based on the Bayesian decision theory, some optimal plug-in estimators are computed.

Keywords Validation of a computer code, Bayesian calibration, Bayesian model selection, Gaussian process emulation, Bayesian decision theory.