Contributions à l'analyse de fiabilité structurale : prise en compte de contraintes de monotonie pour les modèles numériques

Vincent Moutoussamy

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *13/11/2015* **par :**

VINCENT MOUTOUSSAMY

Contributions à l'analyse de fiabilité structurale : prise en compte de contraintes de monotonie pour les modèles numériques

Contributions to structural reliability analysis: accounting for monotonicity constraints in numerical models

---

JURY

Bernard BERCU	Professeur	Univ. Bordeaux I	Examinateur
Nicolas BOUSQUET	Chercheur Expert	EDF R&D	Encadrant industriel
Fabrice GAMBOA	Professeur	Univ. Toulouse III	Directeur de thèse
Bertrand IOOSS	Chercheur Senior	EDF R&D	Encadrant industriel
Thierry KLEIN	Maître De Conférence	Univ. Toulouse III	Directeur de thèse
Sidonie LEFEBVRE	Ingénieur De Recherche	ONERA	Examinatrice
Youssef M. MARZOUK	Associate Professor	MIT	Rapporteur
Bruno TUFFIN	Directeur De Recherche	INRIA Rennes	Rapporteur

École doctorale et spécialité :

MITT : Domaine Mathématiques : Mathématiques appliquées

Remerciements

Mes premiers remerciements vont à Nicolas Bousquet et Bertrand Iooss avec qui j'ai eu la chance de travailler en stage puis en thèse. Je ne pourrais pas vous remercier assez de vos conseils, critiques, disponibilités, encouragements, aides et nombreuses relectures que vous avez fournis pendant ces trois ans et demi.

Je remercie très chaleureusement Fabrice Gamboa et Thierry Klein d'avoir encadré cette thèse. Je vous remercie de votre investissement, votre bienveillance et vos connaissances sans lesquelles cette thèse n'aurait jamais pu aboutir.

Je remercie Youssef Marzouk et Bruno Tuffin d'avoir accepté d'être les rapporteurs de cette thèse. Je les remercie pour leur lecture attentive du manuscrit, leurs remarques ainsi que leur intérêt pour cette thèse. Je remercie également Bernard Bercu et Sidonie Lefebvre d'avoir accepté de faire partie du jury de thèse.

Je remercie Simon Nanty et Benoit Pauwels d'avoir partagé une excellente expérience de recherche lors du CEMRACS 2013. J'en profite pour remercier François Bachoc pour les nombreuses séances de course à pied dans les calanques lors de cette école d'été.

J'ai eu beaucoup de chance de travailler au sein du département MRI. La quantité de compétences différentes a été très stimulante pendant ces trois années. Au-delà des compétences professionnelles, j'ai une pensée toute particulière pour les nombreuses personnes avec qui j'ai pu partager des passions communes. Ne voulant pas prendre le risque d'oublier quelqu'un, j'espère que vous vous reconnaîtrez si j'évoque les anime, manga, dessin, bande-dessinée, jeux vidéo, cinéma, musique, sport, cuisine.

J'ai une pensée pour Thomas Browne, Xavier Yau, Joseph Muré et Nazih Benoumechiara auxquels je souhaite de finir leur thèse sereinement. Mes plus chaleureux remerciements reviennent évidemment à Jeanne Demgne. Comment as-tu fais pour me supporter pendant trois ans dans notre bureau? Un grand merci pour ta bonne humeur, ton rire contagieux et ton écoute!

Je remercie deux amis que j'ai rencontrés lorsque j'étais au lycée. Kevin, que j'ai trouvé un peu effrayant lors de notre première rencontre dans un JDR, dont je ne peux plus me passer de son humour et de ses discussions. Un autre grand merci à Lilian d'avoir organisé à l'époque le "jeudi rhum". J'ai pu rencontrer quelqu'un de passionné, curieux, ouvert et qui a toujours su trouver l'énergie d'être présent lors des moments difficiles!

Je souhaiterais remercier l'ensemble de ma famille pour leur soutien et leurs pensées. En particulier mes parents qui m'ont toujours soutenu dans mes choix. J'ai une pensée très particulière pour ma sœur qui devient, pour son plus grand malheur, mon centre d'attention lorsque l'on se retrouve. Un grand merci à Marie-France et John pour leur gentillesse et d'être venu de Dublin pour encourager leur troisième fils!

Pour finir, je remercie Enora qui malgré son allergie aux mathématiques m'a écouté parler de ma thèse pendant ces trois dernières années. Tu m'as fait grandir et avancer grâce à, je te cite, "des coups de pied au c**". Je n'aurai jamais pu arriver jusqu'ici sans toi. MERCI!

Contents

Notation

g	numerical computer code model
a.s.	almost surely
$\xrightarrow{a.s}$	almost sure convergence
$\xrightarrow{\mathcal{L}}$	convergence in distribution
$\xrightarrow{\mathbb{P}}$	convergence in probability
CV	coefficient of variation
$\|\mathbf{x}\|$	euclidean norm of $\mathbf{x}$
$\|\mathbf{x}\|_q$	q-norm of $\mathbf{x}$
ϕ	Gaussian probability density function
$\mathbf{\Phi}$	Gaussian cumulative distribution function
i.i.d	independent and identically distributed
μ	Lebesgue measure
$\mathbb{P}$	probability
$X, \mathbf{X}$	random variable, vector
$f_{\mathbf{X}}$	probability density function of $\mathbf{X}$
$F_{\mathbf{X}}$	cumulative distribution function of $\mathbf{X}$
$\mathbb{E}[X]$	expectation of X
$Var[X]$	variance of X
$\mathcal{L}(\mathbf{X}\|\mathbf{Y})$	distribution of $\mathbf{X}$ knowing $\mathbf{Y}$
$g \circ f$	function g composed with function h
f^{-1}	inverse function of f
$\preceq$	partial order
$\mathbb{U}^-$	$\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$
$\mathbb{U}^+$	$\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) > 0\}$
Γ	the limit surface $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = 0\}$
$(p^-_{n-1})_{n\geq 1}, (p^+_{n-1})_{n\geq 1}$	deterministic bounds of a probability p
$(q^-_{n-1})_{n\geq 1}, (q^+_{n-1})_{n\geq 1}$	deterministic bounds of a quantile q
$d_H(.,.)$	Hausdorff distance in euclidean norm
$d_{H,q}(.,.)$	Hausdorff distance in q-norm

Résumé étendu

Les installations de production d'EDF, premier producteur d'électricité de France, accueillent des phénomènes physiques pouvant entraîner des dommages aux personnes, à l'environnement et aux biens si la fiabilité structurale de ces installations n'est pas assurée. En France, l'étude de fiabilité des structures s'appuie principalement sur les études de robustesse des installations. Dans des conditions d'exploitation *pénalisantes*, la structure (ou le composant) considérée peut-il encore assurer sa fonction ? Pour certains gros composants industriels dits de haute fiabilité, les données d'exploitation réelle sont uniquement disponibles. Les situations considérées comme pénalisantes n'ont jamais été rencontrées en pratique depuis leur mise en service, et aucune défaillance n'est jamais survenue. C'est notamment le cas de nombreux composants passifs du Parc exploité par EDF, telles que des cuves de production (*vessels* en anglais). En l'absence de données de défaillance, une étude de fiabilité structurale doit nécessairement s'appuyer sur une modélisation mathématique du phénomène. Concrètement, un ou plusieurs modèles numériques sont élaborés et implémentés par les spécialistes de la physique du phénomène, afin de pouvoir le simuler dans les conditions de fonctionnement usuelles et exceptionnelles. On parlera plus naturellement de *code numérique* dans la suite de ce travail. De tels codes résultent souvent de chaînages de codes moins complexes, chacun étant dédié à représenter l'un des phénomènes interagissant sur la dégradation supposée du composant.

Un modèle numérique g permet donc - s'il est validé - d'explorer des configurations critiques, incarnées par le choix de ses paramètres d'entrée $\mathbf{x}$[1], et de déterminer si ces configurations engendrent un risque de défaillance. Celle-ci peut très grossièrement être définie ainsi : "la contrainte appliquée $C(\mathbf{x})$ [2] est égale ou supérieure à la résistance $R(\mathbf{x})$ de la structure (ou du composant)". Ainsi, en définissant génériquement $g(\mathbf{x}) = R(\mathbf{x}) - C(\mathbf{x})$, une configuration sera considérée comme menant à une défaillance si $g(\mathbf{x}) \leq 0$. Outre la vérification que certaines configurations pénalisantes n'aboutissent pas à des situations de défaillance, les études de sûreté industrielle emploient couramment deux indicateurs complémentaires :

- la probabilité d'occurence d'une défaillance $\mathbb{P}(g(\mathbf{X}) \leq 0)$, en supposant que les paramètres d'entrée $\mathbf{X}$, souvent connus avec une certaine incertitude, peuvent varier aléatoirement dans des plages réalistes ;
- l'indicateur *dual* défini comme la différence entre résistance et contrainte Z_α tel que $\mathbb{P}(g(\mathbf{X}) \leq Z_\alpha)$ reste en dessous d'un *seuil d'acceptabilité* α fixé par une règle de sûreté.

Le calcul de ces deux indicateurs permet en parallèle de résoudre (partiellement) le problème *inverse* consistant à déterminer les configurations d'entrée menant à des situations de défaillance.

[1] qui mélangent des paramètres de fonctionnement (ex : pression), des forçages environnementaux (ex : température) et, entre autres, des caractéristiques matériaux.

[2] Par exemple, une injection d'eau de refroidissement dans une cuve en acier portée à haute température.

Dans la pratique, de tels codes complexes sont considérés comme des "boîtes noires", ce qui signifie qu'on ne peut accéder aux détails des équations qu'ils implémentent dans un temps raisonnable. Ainsi, des techniques d'exploration dites *intrusives*, notamment fondées sur la différentiabilité de ces codes, sont difficiles voire impossibles à mettre en oeuvre. L'exploration des codes, et le calcul des indicateurs fiabilistes en particulier, doit être réalisée en pratique par des techniques non-intrusives, fondées sur des parcours de simulation.

Les méthodes de type Monte Carlo, naturelles dès lors qu'on traite de simulation, permettent en théorie d'obtenir des estimateurs statistiques de ces indicateurs fiabilistes. Toutefois, un coût de simulation (en temps ou espace mémoire) important est souvent consubstantiel à la complexité et la précision du code, qui peut en pratique interdire l'usage de certaines de ces méthodes. C'est pourquoi un vaste champ de techniques, dites de Monte Carlo accélérées (ou de réduction de variance), connaît un essor important depuis plusieurs années. Il vise à produire des plans d'expériences de simulation numérique *intelligents*, permettant de telles estimations à faible coût computationnel. Ce travail de thèse se situe explicitement dans ce champ d'étude.

La vision "boîte noire" des codes n'est pas, à proprement parler, tout à fait exacte dans la réalité des études de fiabilité. En réalité, la notion de *configuration pénalisante* implique nécessairement celle de *monotonie globale du phénomène* : plus un stress (contrainte) augmente, plus le risque croît. Réciproquement, plus une résistance augmente, plus le risque décroît. Il apparaît donc raisonnable de supposer qu'une configuration moins contraignante qu'une configuration sûre mène aussi à un événement sûr. Dans de nombreux cas, cette monotonie décrit le comportement du phénomène - et donc du code correspondant, supposé valide - vis-à-vis de ses variables les plus influentes. Du point de vue des études de sûreté, l'avantage de la monotonie est *a priori* considérable, car elle permet d'encadrer les valeurs des indicateurs de fiabilité de façon déterministe, et non plus seulement probabiliste (via un intervalle de confiance).

Ce travail de thèse fait donc plutôt l'hypothèse que le code peut être considéré comme une "boîte grise", dont la monotonie en fonction des entrées incertaines est connue. Certains travaux parallèles sont en cours à EDF R&D pour vérifier la réalité de cette hypothèse, mais ils ne sont pas intégrés dans ce document. Ce travail suppose également que les lois probabilistes des paramètres d'entrée sont connues, et indépendantes. Cette hypothèse d'indépendance peut être relâchée sous certaines conditions, mais qui ne font pas l'objet d'un travail spécifique dans cette thèse. Par ailleurs, très peu d'hypothèses supplémentaires sont faites sur le code : celui-ci, en particulier, peut représenter une physique discontinue (parfois typique des configurations critiques[3]) et présenter des *effets-falaises*. Enfin, la sortie de ce code peut, dans certains cas, être réduite à la simple expression binaire :

$$Z(\mathbf{x}) = \mathbb{1}_{\{g(\mathbf{x})\leq 0\}},$$

ce qui permet de considérer ce code comme un outil d'aide à la décision au sens large (et non plus la "simple" représentation d'un phénomène physique). Dans un tel contexte, d'ailleurs, la notion d'ordre partiel sur des décisions apparaît comme essentielle car elle génère de la monotonie, qui permet de préférer une décision à une autre. Cette notion d'ordre partiel sera au coeur des travaux effectués dans cette thèse.

Ce travail porte donc sur l'utilisation des propriétés de monotonie des codes de calcul pour l'amélioration de l'estimation des indicateurs fiabilistes, dans un contexte où il est souhaitable

[3]Par exemple, la physique d'un cours d'eau peut être discontinue à l'approche de la crue, caractérisée par une surverse au-dessus d'une digue.

que le nombre de simulations à effectuer soit le plus faible possible (pour une précision donnée) ou respecte un budget prédéfini. Cette amélioration est à comprendre au sens de la précision des bornes déterministes évoquées précédemment, mais aussi de la vitesse de convergence des estimateurs statistiques situés entre ces bornes.

Plus précisément, trois problèmes ont été étudiés dans cette thèse.

1. Le comportement théorique des bornes, qui constituent des objets probabilistes du fait qu'elles sont construites à partir de plans d'expériences numériques *simulés*. L'étude des plans d'expérience permettant une bonne convergence de ces bornes en constitue un aspect. Cette étude a enfin mené à construire et étudier théoriquement un *méta-modèle* de surface de l'état-limite $\{\mathbf{x}, g(\mathbf{x}) = 0\}$ (surface de défaillance).

2. L'estimation accélérée de la probabilité de défaillance définie ci-dessus, notamment par l'élaboration de plans d'expériences séquentiels.

3. L'estimation accélérée du quantile également défini, par des procédés similaires.

Enfin, un cas d'étude traitant de la fiabilité de certains composants de production d'EDF, dont l'étude constituait la motivation industrielle de cette thèse CIFRE, est traité dans un chapitre dédié.

Les principaux résultats de recherche obtenus au cours de la thèse sont repris plus formellement dans les sections suivantes.

Apport de la monotonie pour l'estimation de probabilité de dépassement de seuil (Chapitre 2)

L'hypothèse centrale de la thèse porte sur la monotonie d'un code. Cette hypothèse est formalisée dans la définition suivante.

Définition 2.1. La fonction g est dite globalement monotone si elle est monotone relativement à chacune de ses entrées.

Chacune des entrées du code va avoir un impact favorable ou défavorable sur la fiabilité. Pour simplifier la construction, on suppose sans perte de généralité que g est globalement croissante. Ce genre de transformation est courante en fiabilité structurale (voir sections 1.6 et 1.7). La probabilité que l'on cherche à estimer est définie par

$$p = \mathbb{P}(g(\mathbf{X}) \leq q),$$

avec $\mathbf{X}$ un vecteur aléatoire dont les composantes sont indépendantes entre elles. La transformation de g permet de considérer que $\mathbf{X}$ est un vecteur aléatoire uniformément distribué sur $[0,1]^d$ (voir section 2.2). Il faut remarquer que faire cette transformation nécessite de connaître la monotonie de g selon chacune de ses entrées. Enfin, par une translation, on considère sans perte de généralité que $q = 0$. On définit un ordre partiel sur $\mathbb{R}^d$.

Définition 2.2. Soit $\mathbf{x} = (x_1, \dots, x_d), \mathbf{y} = (y_1, \dots, y_d)$ deux points de $\mathbb{R}^d$ tels que pour tout $i = 1, \dots, d,\ x_i \leq y_i$. Cette relation est un ordre partiel et est notée $\mathbf{x} \preceq \mathbf{y}$. On dit aussi que $\mathbf{y}$ domine $\mathbf{x}$.

Cela signifie que pour tout $\mathbf{x}, \mathbf{y} \in [0,1]^d$ tels que $\mathbf{x} \preceq \mathbf{y}$ alors $g(\mathbf{x}) \leq g(\mathbf{y})$. Si on sait que $\mathbf{y}$ mène à un événement indésirable alors $\mathbf{x}$ mène aussi à l'événement redouté. Cette information est obtenue sans évaluer g en $\mathbf{x}$. Pour simplifier la lecture, on note

$$\begin{aligned}\mathbb{U}^- &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) \leq 0\},\\ \mathbb{U}^+ &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) > 0\},\\ \Gamma &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = 0\}.\end{aligned}$$

Ces ensembles représentent l'ensemble des configurations menant respectivement à un événement sûr, à un événement indésirable et à la surface limite séparant ces deux ensembles. Soit A un ensemble de $[0,1]^d$. On note

$$\begin{aligned}\mathbb{U}^-(A) &= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^-} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \preceq \mathbf{x}\},\\ \mathbb{U}^+(A) &= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^+} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \succeq \mathbf{x}\},\end{aligned}$$

avec $\mathbb{U}^-(\emptyset) = \{0\}^d = (0, \dots, 0)$ et $\mathbb{U}^+(\emptyset) = \{1\}^d = (1, \dots, 1)$. Tout l'intérêt de la monotonie pour l'estimation de probabilité se résume aux deux équations suivantes :

$$\begin{aligned}\mathbb{U}^-(A) \subset \mathbb{U}^- \subset [0,1]^d \backslash \mathbb{U}^+(A),\\ \mu(\mathbb{U}^-(A)) \leq p \leq 1 - \mu(\mathbb{U}^+(A)),\end{aligned}$$

avec μ la mesure de Lebesgue sur $\mathbb{R}^d$. On peut donc encadrer à la fois l'état limite Γ et la probabilité p. Comme le nombre d'appels au code est limité, l'ensemble A prend la forme d'un ensemble de points. Les prochaines notations vont revenir régulièrement dans cette thèse. Soit $(\mathbf{X}_k)_{k \geq 1}$ une suite de points ou de vecteurs aléatoires. On notera continuellement s'il n'y a pas d'ambiguïté,

$$\begin{aligned}\mathbb{U}_n^- &= \mathbb{U}^-(\mathbf{X}_1, \dots, \mathbf{X}_n),\\ \mathbb{U}_n^+ &= \mathbb{U}^+(\mathbf{X}_1, \dots, \mathbf{X}_n),\\ \mathbb{U}_n &= [0,1]^d \backslash (\mathbb{U}_n^- \cup \mathbb{U}_n^+),\\ p_n^- &= \mu(\mathbb{U}_n^-),\\ p_n^+ &= 1 - \mu(\mathbb{U}_n^+).\end{aligned}$$

La figure 1 illustre cette construction avec A un ensemble de points. La mesure des ensembles représentés en gris fournissent des bornes pour p.

L'hypothèse de monotonie permet de connaître le signe de g en certains points grâce aux évaluations précédentes. Cela s'exploite facilement avec l'estimateur de Monte Carlo classique. Soit $(\mathbf{X}_k)_{k \geq 1}$ une suite de vecteurs aléatoires indépendant et de loi uniforme sur $[0,1]^d$. Si on se donne un budget de n évaluations au code numérique, l'estimateur de Monte Carlo de p est

$$\hat{p}_n^{MC} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{X}_i) \leq 0\}}. \tag{1.5}$$

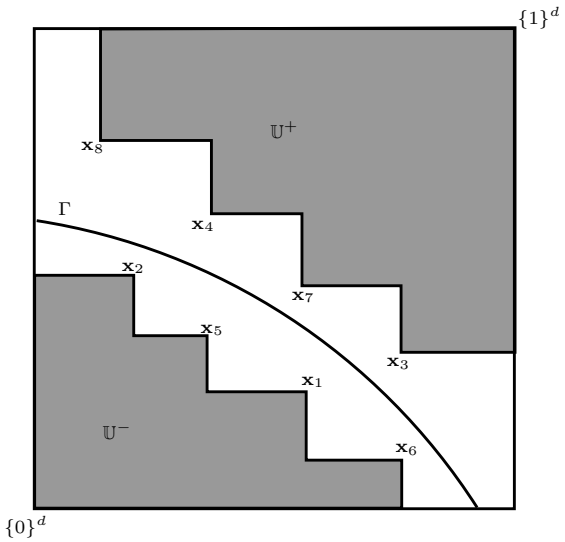


Figure 1: Illustration pour $d = 2$ de $\mathbb{U}^-(A)$ et $\mathbb{U}^+(A)$ avec $A = \{\mathbf{x}_1, \ldots, \mathbf{x}_8\}$.

Supposons que $g(\mathbf{X}_1) > 0$. Si $\mathbf{X}_2$ domine $\mathbf{X}_1$ alors sans faire d'appel à g on sait que $g(\mathbf{X}_2) > 0$. Le signe de g est connu en deux points alors qu'une seule évaluation a été faite. En prenant en compte cette construction, p peut s'estimer par

$$\frac{1}{N_n} \sum_{k=1}^{N_n} \mathbb{1}_{\{g(\mathbf{X}_k) \leq 0\}}, \tag{2.5}$$

où N_n est le nombre de simulations faites avec un budget de n appels au code. La valeur N_n est aléatoire. En effet, à chaque étape k la probabilité d'économiser un appel au code vaut $1 - \mathbb{E}[p_{k-1}^+ - p_{k-1}^-]$. Néanmoins, les bornes de p n'interviennent pas directement dans la construction de cet estimateur.

Une fois les bornes obtenues, la probabilité p peut se réécrire comme

$$p = p_n^- + (p_n^+ - p_n^-)\mathbb{P}(\mathbf{X} \in \mathbb{U}^- | \mathbf{X} \in \mathbb{U}_n).$$

Cette écriture est proche de celle obtenue par la méthode de *simulation multi-niveau* décrite dans la section 1.8. Si les bornes sont connues, p peut être estimée par

$$\widehat{p}_N = p_n^- + \frac{p_n^+ - p_n^-}{N} \sum_{k=1}^{N} \mathbb{1}_{\{\mathbf{X}_k^{(n)} \in \mathbb{U}^-\}}, \tag{2.7}$$

où $(\mathbf{X}_k^{(n)})_{k \geq 1}$ est une suite de vecteurs aléatoire indépendant et uniformément distribués sur $\mathbb{U}_{n-1}$. Cette stratégie n'est toujours pas optimale. La connaissance des indicatrices présentes dans la somme n'est pas exploitée pour mettre à jour les bornes de p.

Finalement, il semble nécessaire de simuler, à chaque étape n, sur l'ensemble *non-dominé* $[0,1]^d \backslash (\mathbb{U}_{n-1}^- \cap \mathbb{U}_{n-1}^+)$. De cette manière les bornes et l'ensemble non-dominé sont mis à jour à chaque évaluation du code numérique.

On peut aussi utiliser des critères pour choisir un nouveau point à évaluer. Mais il semble difficile d'utiliser un critère pour calibrer une fonction d'importance.

Étude théorique du comportement des bornes (Chapitre 3)

Les bornes déterministes obtenues sont utiles pour construire des estimateurs efficaces de p. Néanmoins, il faut s'assurer qu'elle convergent bien vers p. En plus de la convergence des bornes, la convergence de la suite d'ensembles $(\mathbb{U}_k^-)_{k\geq 1}$ a été étudiée. La distance de Hausdorff, définie ci-dessous, s'avère utile pour comparer deux ensembles.

Définition 3.2. Soit $\|\cdot\|_q$ la norme $\mathcal{L}^q$ sur $\mathbb{R}^d$ définie pour $0 < q < +\infty$, $\|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}$, et pour $q = +\infty$, $\|\mathbf{x}\|_\infty = \max_{i=1,\dots,d_{i=1,\dots,d}} x_i$. Soit (A, B) deux ensembles non vides de l'espace vectoriel normé $([0,1]^d, \|\cdot\|_q)$. La distance de Hausdorff $d_{H,q}$ est définie par

$$d_{H,q}(A,B) = \max(\sup_{\mathbf{y}\in A}\inf_{\mathbf{x}\in B}\|\mathbf{x}-\mathbf{y}\|_q;\ \sup_{\mathbf{x}\in B}\inf_{\mathbf{y}\in A}\|\mathbf{x}-\mathbf{y}\|_q).$$

Une façon naïve d'y arriver est de simuler sur la surface limite Γ. Évidement cela est impossible en pratique mais donne une piste pour résoudre ce problème. Les résultats théoriques montrent qu'il faut simuler infiniment souvent autour de Γ (voir proposition 3.2). Les deux schémas de simulations étudiés sont l'échantillonnage de Monte Carlo et la simulation uniforme dans l'espace non-dominé. L'utilisation de ces deux schémas assurent la convergence des bornes ainsi que la convergence des ensembles $\mathbb{U}_n^-$ et $\mathbb{U}_n^+$ respectivement vers $\mathbb{U}^-$ et $\mathbb{U}^+$.

On s'intéresse à la vitesse de convergence des bornes issues de ces deux stratégies de simulation. Sous certaines conditions de régularités (voir définition 3.3) on peut connaître la vitesse de convergence de la suite $(\mathbb{U}_{n-1}^-)_{n\geq 1}$ vers $\mathbb{U}^-$.

Proposition 3.4. Soit $(\mathbf{X}_k)_{k\geq 1}$ une suite de vecteurs aléatoires indépendants et uniformément distribués sur $[0,1]^d$ et $(\tilde{\mathbf{X}}_k)_{k\geq 1}$ une suite de vecteurs aléatoires indépendants et uniformément distribués sur $\mathbb{U}^-$. On note $\tilde{\mathbb{U}}_n^- = \mathbb{U}^-(\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_n)$. Soit $(F_n)_{n\geq 1}$ une suite de sous-ensemble mesurables de $[0,1]^d$ tel que pour tout $n \geq 1, \mathbb{U}_n^- \subset F_n \subset [0,1]^d \backslash \mathbb{U}_n^+$. Alors

(1) $d_{H,2}(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{p.s.} 0$ et $\mu(F_n) \xrightarrow[n\to+\infty]{p.s.} p$.

(2) If $\mathbb{U}^-$ est régulier, alors presque sûrement

$$d_{H,2}(\tilde{\mathbb{U}}_n^-, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

(3) De plus, si $\mathbb{U}^+$ est aussi régulier, et si g est continue, alors presque sûrement

$$d_{H,2}(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

Pour la vitesse des bornes, on commence par le cas $d = 1$ avec un schéma de simulation indépendant et uniformément distribué sur $[0,1]$. La proposition 3.5 dit que

$$\begin{aligned} n(p - p_n^-) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1), \\ n(p_n^+ - p) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1), \\ \mathbb{E}[p_n^+ - p_n^-] &= \frac{2}{n+1} - \frac{1}{n+1}\left(p^{n+1} + (1-p)^{n+1}\right), \end{aligned}$$

où $\mathcal{E}xp(\lambda)$ est la loi exponentielle de paramètre λ et de fonction de densité $f_\lambda(x) = \lambda e^{-\lambda x}\mathbb{1}_{\{x\geq 0\}}$.

Une simulation uniforme n'est pas optimale car elle n'exploite pas la connaissance des bornes. De plus, il va devenir de plus en plus difficile de les mettre à jour lorsque n devient grand. En effet, la probabilité que X_n mette à jour les bornes est de l'ordre de $1/n$. Il faudra donc faire de plus en plus de simulations avant de pouvoir obtenir de nouvelles bornes. Pour accélérer la vitesse de convergence il aurait été plus judicieux de simuler à chaque étape n sur l'intervalle $]p_{n-1}^-, p_{n-1}^+[$. La proposition 3.6 donne l'écart moyen des bornes pour une telle stratégie :

$$\frac{1}{2^n} \leq \mathbb{E}[p_n^+ - p_n^-] \leq \left(\frac{3}{4}\right)^n.$$

Comme supposé, la prise en compte de l'information fournie par les bornes accélère significativement la vitesse de convergence. Mais le cas $d = 1$ a peu d'intérêt en pratique car cela revient à la recherche du zéro d'une fonction croissante. Il est donc intéressant de comprendre l'influence de la dimension sur ces quantités. On considère maintenant un cas plus général en supposant que $d \geq 2$. Pour $d = 1$ la surface limite Γ est unique et est égale à $\{p\}$. Mais cela n'est plus valide en dimension supérieure. Il est donc difficile d'obtenir les même résultats qu'en dimension un pour une valeur de p quelconque. On se restreint à la valeur $p = 1$. La proposition 3.7 fournit un résultat similaire à la proposition 3.6 pour la suite $(\mathbb{U}_k^-)_{k\geq 1}$. En effet, pour $d = 1$, $d_{H,q}(\mathbb{U}_n^-, [0,1]) = 1 - p_{n-1}^-$.

Proposition 3.7. Supposons que $\Gamma = \{1\}^d$. Soit $(\mathbf{X}_k)_{k\geq 1}$ une suite de vecteurs aléatoires indépendants et uniformément distribués sur $[0,1]^d$. Pour $0 < q < +\infty$ on note $A(1,q) = 1$ et pour $d \geq 2$, $A_{d,q} = \frac{1}{dq^{d-1}} \prod_{i=1}^{d-1} B(i/q, 1/q)$ avec $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$. Pour tout $n \geq 1$, on note $\mathbb{U}_n^- = \mathbb{U}^-(\mathbf{X}_1, \ldots, \mathbf{X}_n)$.

(1) Si $0 < q < +\infty$ alors

$$(A_{d,q}n)^{1/d} d_{H,q}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d).$$

(2) Si $q = +\infty$ alors

$$n^{1/d} d_{H,\infty}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d),$$

où $\mathcal{W}(1,d)$ est la loi de Weibull avec un paramètre d'échelle de 1 et un paramètre de forme d ayant pour fonction de répartition $F(t) = 1 - e^{-t^d}$ for all $t \geq 0$.

La proposition 3.8 donne un ordre de grandeur de l'écart moyen entre les bornes:

$$\mathbb{E}[1 - p_n^-] \underset{n\to+\infty}{\sim} \frac{\log(n)^{d-1}}{n(d-1)!}.$$

Exploiter la monotonie devient moins intéressant lorsque la dimension augmente. Lorsque l'on rajoute des paramètres d'entrée à un code numérique, ce code peut ne plus être globalement monotone. L'hypothèse de monotonie est donc plutôt adaptée aux petites dimensions.

Un schéma de simulation séquentielle semble indispensable pour exploiter au mieux la monotonie d'un code. Lorsque l'on veut estimer Γ, une approche séquentielle est aussi plus intéressante. En supposant que Γ soit convexe (ou concave), il est possible de construire un estimateur adaptatif de Γ. Celui-ci est élaboré à partir de plusieurs classifieurs linéaires produits par *Machine à Vecteur Support* (SVM). Chacun de ces classifieurs linéaires est obtenu à partir de

certains points d'un échantillon. L'aspect adaptatif vient que pour chaque nouvelle simulation, un nouveau classifieur linéaire est ajouté à l'estimateur de Γ. Il n'est pas forcément nécessaire de tout reconstruire. Lorsque le temps de calcul est facteur important, cette construction séquentielle apporte un intérêt concret. On décrit maintenant la construction de cet estimateur.

On se donne un plan d'expérience $D_n = (\mathbf{X}_i, y_i)_{1\leq i\leq n} \in [0,1]^d \times \{-1,1\}$ où $y_i = 1$ si $g(\mathbf{X}_i) > 0$ et -1 sinon. Soit $\Xi_n^+ = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^+$ et $\Xi_n^- = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^-$ et pour tout $\mathbf{x} \in \Xi_n^+$ on définit $h_{\mathbf{x}}$ comme l'hyperplan qui sépare $\mathbf{x}$ de Ξ_n^-. Un classifieur f_n est défini par

$$\begin{aligned} f_n : [0,1]^d &\to \{-1,+1\} \\ \mathbf{y} &\mapsto \begin{cases} -1 & \text{si pour tout } \mathbf{X} \in \Xi_n^+,\ h_{\mathbf{X}}(\mathbf{y}) \leq 0 \\ +1 & \text{sinon,} \end{cases} \end{aligned}$$

et on note

$$F_n = \{\mathbf{x} \in [0,1]^d,\ f_n(\mathbf{x}) = -1\},$$

un estimateur de $\mathbb{U}^-$. La construction de cet estimateur est illustrée sur la figure 2 en dimension $d = 2$. Le théorème 3.1 donne les principales propriétés de f_n et F_n.

Théorème 3.1. On suppose que $\mathbb{U}^-$ est convexe, alors

(1) f_n est globalement croissante.

(2) Pour tout $\mathbf{X} \in \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, $sign(g(\mathbf{X})) = f_n(\mathbf{X})$.

(3) L'ensemble F_n est un polyèdre convexe.

(4) De plus, si $(\mathbf{X}_k)_{k\geq 1}$ est une suite indépendantes de vecteurs aléatoires uniformément distribués sur $[0,1]^d$, alors

$$d_{H,2}(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{p.s.} 0,$$

et presque sûrement

$$d_{H,2}(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

Estimation de probabilité (Chapitre 4)

Dans ce chapitre, on veut pouvoir fournir un estimateur de la probabilité p. Cet estimateur doit avoir de meilleures propriétés que celui construit dans [16] qui se fonde déjà sur l'hypothèse de monotonie. Il ne doit pas avoir de biais, avoir une variance plus faible, réduire la borne supérieure de la probabilité cherchée et enfin être facilement utilisable. L'équation (4.6) fournit une écriture général d'un estimateur non biaisé de p

$$\widetilde{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^- + \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^-\}} \right), \tag{4.6}$$

où $\omega_{1,n}, \ldots, \omega_{n,n}$ est une suite de réels positifs tels que $\sum_{k=1}^n \omega_{k,n} = 1$. L'estimateur $\widetilde{p}_n$ est sans biais si pour tout $k \geq 1$ et pour tout $\mathbf{x} \in \mathbb{U}_{k-1} \cap \mathbb{U}^-$, $f_{k-1}(\mathbf{x}) > 0$. Des résultats classiques en simulation préférentielle fournissent la suite de fonctions de densité $(f_{k-1})_{k\geq 1}$ qui annule la variance de $\widetilde{p}_n$:

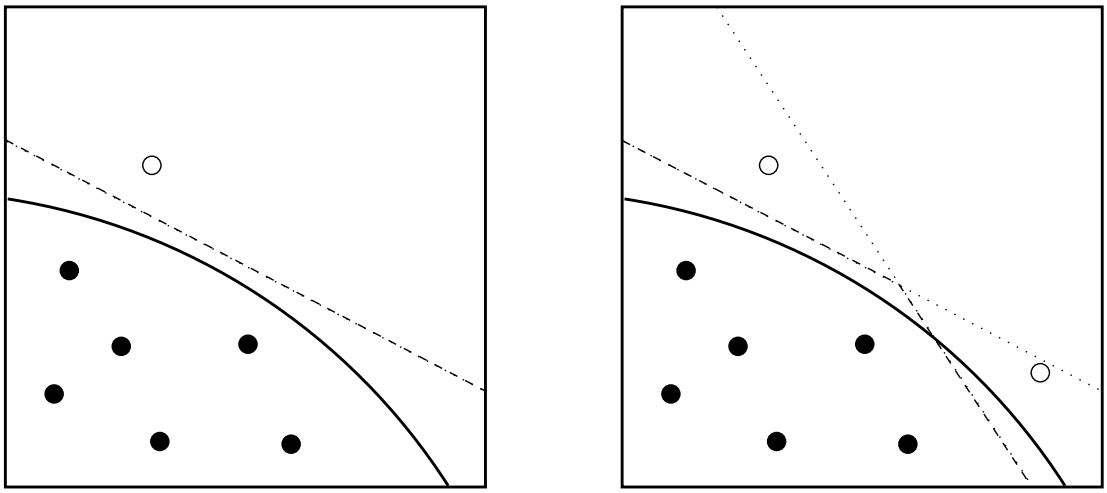


Figure 2: Construction du classifieur fondé sur les SVM en dimension 2. La courbe représente Γ et les points noirs (resp. blancs) sont dans $\mathbb{U}^-$ (resp. $\mathbb{U}^+$). La ligne pointillé représente $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ pour $\mathbf{X}$ dans $\mathbb{U}^+$. **Gauche** : il n'y a qu'un seul point dans $\mathbb{U}^+$, la ligne en tiret représente $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ et la frontière du classifieur. **Droite** : les lignes pointillés représentent les ensembles $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ pour $\mathbf{X}$ sélectionné comme l'un deux points blancs. La ligne en tiret représente la frontière du classifieur.

$$f_{k-1}(\mathbf{x}) = \frac{\mathbb{1}_{\{\mathbf{x} \in \mathbb{U}^- \cap \mathbb{U}_{k-1}\}}}{p - p_{k-1}^-}. \tag{4.7}$$

Évidemment cette densité optimale ne peut pas être utilisée en pratique mais elle donne une piste de construction. Elle indique qu'il faut simuler uniformément sur $\mathbb{U}^- \cap \mathbb{U}_{k-1}$. Cela implique que seule la borne inférieure sera mise à jour. Comme l'un des objectifs est de réduire la borne supérieure de la probabilité cherchée, on peut construire l'estimateur suivant :

$$\widehat{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^+ - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}} \right). \tag{4.8}$$

Cette fois ci, la variance est minimum si, à chaque étape k, on simule uniformément sur $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$. Comme précédemment cela ne peut pas être fait en pratique. On cherche donc à s'approcher de la densité optimale tout en gardant la propriété que l'estimateur soit sans biais. On considère à l'étape k un estimateur $\widehat{\mathbb{U}}_{k-1}^+$ de $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$. S'il est suffisamment précis, une simulation uniforme sur cet ensemble sera proche de la distribution optimale. Mais comme il est impossible de vérifier que $\widehat{\mathbb{U}}_{k-1}^+ \supset \mathbb{U}^+ \cap \mathbb{U}_{k-1}$, il n'est pas certain que l'estimateur produit soit sans biais. On propose alors de simuler $\mathbf{X}_k$ de la manière suivante

$$\mathbf{X}_{k-1} \sim \begin{cases} \mathcal{U}(\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}_{k-1}^+) & \text{avec probabilité } \varepsilon_{k-1}, \\ \mathcal{U}(\widehat{\mathbb{U}}_{k-1}^+) & \text{avec probabilité } 1 - \varepsilon_{k-1}, \end{cases} \tag{4.10}$$

Le choix de ε_{k-1} est crucial pour minimiser la variance de l'estimateur obtenu. Mais ce choix de ε_{k-1} implique de connaître $\mathbb{U}^+$ qui est inconnu. Comme $\mathbb{U}^+$ doit être estimé à chaque itération, l'utilisation de telles méthodes est très coûteuse en temps de calcul. Finalement, le cas simple suivant a été étudié. Soit $(\mathbf{X}_k)_{k \geq 1}$ une suite de vecteurs aléatoire tels que pour tout $k \geq 1, \mathbf{X}_k$ est uniformément distribué sur $\mathbb{U}_{k-1}$. On note

$$\bar{p}_k = p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-) \mathbb{1}_{\{g(\mathbf{X}_k) \leq 0\}},$$

et un estimateur sans biais de p devient

$$\widehat{p}_n = \frac{1}{n}\sum_{k=1}^{n} \bar{p}_k.$$

Il faut remarquer que la simulation uniforme sur l'ensemble non-dominé n'est pas si éloigné de la densité optimale. En effet, simuler uniformément sur l'ensemble non-dominé revient à choisir $\widehat{\mathbb{U}}^+_{k-1} = \mathbb{U}^+ \cap \mathbb{U}_{k-1}$. Les tests numériques obtenus sur un exemple montrent que $\mathbb{E}[1-\varepsilon_{k-1}] = \mathbb{E}[(p^+_{k-1}-p)/(p^+_{k-1}-p^-_{k-1})]$ est proche de 1. C'est à dire qu'avec grande probabilité les simulations sont faites sur $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$.

On a comparé la vitesse de convergence de la borne supérieure pour deux schémas de simulations. Le premier est la simulation uniforme sur l'ensemble non-dominé et le second selon la densité optimale. Comme on l'a dit précédemment cette densité n'est pas utilisable en pratique. Mais avec une méthode de rejet on a pu simuler uniformément sur $\mathbb{U}^+$. Les expériences numériques ont montré que l'utilisation de la densité optimale ne réduit pas significativement la borne supérieure par rapport à une simulation uniforme sur $\mathbb{U}_{k-1}$.

Finalement, la simulation uniforme semble plus pratique à utiliser et permet d'avoir un estimateur sans biais. La simulation optimale ne semble pas être significativement plus efficace pour réduire la borne supérieure. De plus, cela ne demande pas de construire à chaque étape un estimateur de $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$. L'ensemble de ces remarques indique que l'estimateur produit a de bonnes propriétés et est facilement utilisable en pratique.

Cependant cet estimateur reste dans l'état actuel difficilement contrôlable. En effet, les outils théoriques disponibles montrent que l'estimateur a une variance trop faible. Il ne varie pas assez pour que l'on puisse obtenir un théorème limite central.

Estimation de quantile (Chapitre 5)

L'objectif de ce chapitre est de fournir un estimateur consistant ainsi qu'un encadrement sûr à 100% d'un quantile. Pour encadrer une probabilité il suffit de connaître le signe de g sur un ensemble de points. Cette approche n'est pas viable pour encadrer un quantile. Soit F la fonction de répartition de $g(\mathbf{X})$. Le quantile d'ordre p de $g(\mathbf{X})$ est défini par

$$q = \inf\{t \in \mathbb{R},\ F(t) \geq p\}. \tag{5.1}$$

On donne quelques définitions pour simplifier la présentation.

Définition 5.1. Soit $A \subset [0,1]^d$. On définit

$$\mathbb{V}^-(A) = \bigcup_{\mathbf{x}\in A} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \preceq \mathbf{x}\},$$
$$\mathbb{V}^+(A) = \bigcup_{\mathbf{x}\in A} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \succeq \mathbf{x}\}.$$

Définition 5.2. Soit $\alpha \in]0,1[$ et S un ensemble de $[0,1]^d$. On dit que S est α-monotone si pour tout $\mathbf{u}, \mathbf{v} \in S$, $\mathbf{u}$ n'est pas strictement dominé par $\mathbf{v}$ et si $\mu(\mathbb{V}^-(S)) = \alpha$.

L'idée pour obtenir un tel encadrement vient entièrement de la proposition suivante.

Proposition 5.3. Pour tout $\alpha \in]0,1[$ on suppose S_α est α-monotone. Soit $p^- \in [0,p[$ et $p^+ \in [p,1[$, alors

$$\min_{\mathbf{x}\in S_{p^-}} g(\mathbf{x}) \leq q \leq \max_{\mathbf{x}\in S_{p^+}} g(\mathbf{x}).$$

On remarque que $\Gamma = \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\}$ est p-monotone. En pratique il est difficile de construire de tels ensembles. Cela peut nécessiter de faire trop d'appels au code pour trouver le minimum et le maximum de g sur cet ensemble. Deux contraintes pratiques sont à prendre en compte. La première est de pouvoir construire facilement un ensemble monotone. La seconde est d'avoir à faire un nombre limité d'appels à g pour trouver le minimum et le maximum. La solution qui a été trouvée est de construire un ensemble monotone à partir d'un ensemble de points. Soit $\mathbf{x} = (x^1, \ldots, x^d)$ un point de $[0,1]^d$. La frontière de $\mathbb{V}^-(\mathbf{x})$ est monotone. Comme g est globalement croissant, son maximum sur la frontière de $\mathbb{V}^-(\mathbf{x})$ est atteint en $\mathbf{x}$. Si $\mu(\mathbb{V}^-(\mathbf{x})) = x^1 \cdots x^d \geq p$ alors la contrainte de volume est vérifiée et $g(\mathbf{x}) \geq q$. Une construction similaire permet d'obtenir une borne inférieure pour q. Ces constructions mènent à la proposition 5.4.

Proposition 5.4. Soit $p \in]0,1[$ et $d \geq 2$. On note

$$\mathbb{W}^-(p) = \left\{\mathbf{u} = (u^1, \ldots, u^d) \in [0,1]^d, \prod_{i=1}^d (1-u^i) \geq 1-p\right\},$$
$$\mathbb{W}^+(p) = \left\{\mathbf{u} = (u^1, \ldots, u^d) \in [0,1]^d, \prod_{i=1}^d u^i \geq p\right\}.$$

Pour tout $(\mathbf{u}, \mathbf{v}) \in \mathbb{W}^-(p) \times \mathbb{W}^+(p)$ il vient que

$$g(\mathbf{u}) \leq q \leq g(\mathbf{v}).$$

Soit $\mathbb{W}(p) = [0,1]^d \backslash (\mathbb{W}^-(p) \cup \mathbb{W}^+(p))$, alors

$$\mu(\mathbb{W}(p)) = (1-p)\sum_{k=0}^{d-1}\left[\frac{(-\log(1-p))^k}{k!}\right] + p\sum_{k=0}^{d-1}\left[\frac{(-\log(p))^k}{k!}\right] - 1.$$

L'ensemble $\mathbb{W}(p)$ représente l'ensemble des points de $[0,1]^d$ où l'on ne sait pas si g est plus petit ou plus grand que q. Cette construction est universelle car elle ne dépend pas de g. Pour toute fonction g globalement croissante on sait que $\{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\} \subset \mathbb{W}(p)$ où q est le p-quantile de $g(\mathbf{X})$.

Cette étape d'initialisation est particulièrement intéressante lorsque p tend vers 0 ou 1. En effet, $\mu(\mathbb{W}(p))$ tend vers 0 lorsque p tend vers 0 ou 1. On a représenté sur la figure 3 l'ensemble $\mathbb{W}(p)$ en dimension 2 et pour différentes valeurs de p.

Maintenant qu'une étape d'initialisation est faite, on veut pouvoir encadrer q à partir d'un ensemble de points E. La proposition 5.3 dit que

$$\mu(\mathbb{V}^-(E)) > p \Rightarrow \max_{\mathbf{x}\in E} g(\mathbf{x}) \geq q,$$
$$\mu(\mathbb{V}^+(E)) > 1-p \Rightarrow \min_{\mathbf{x}\in E} g(\mathbf{x}) \leq q.$$

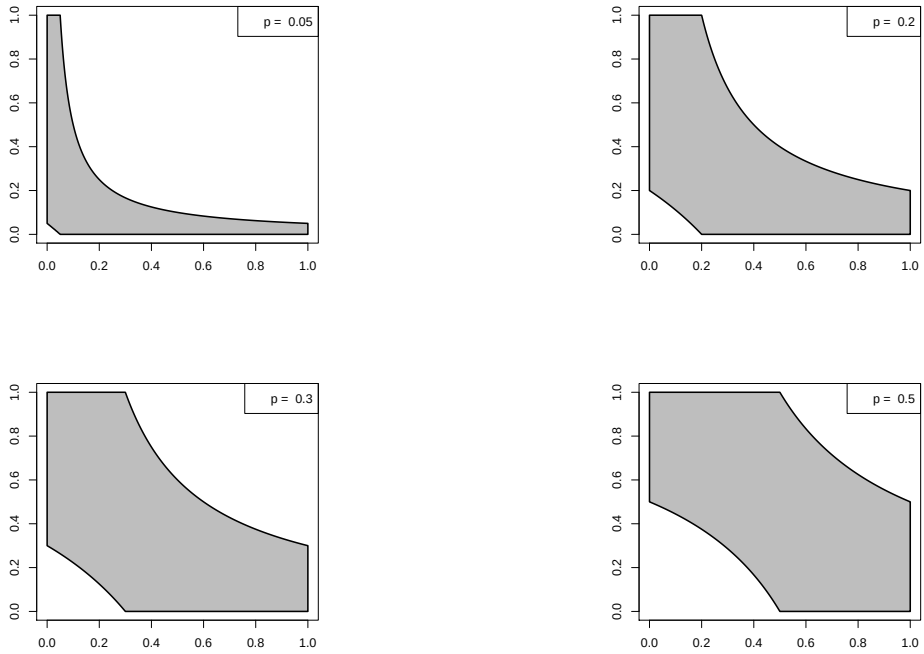


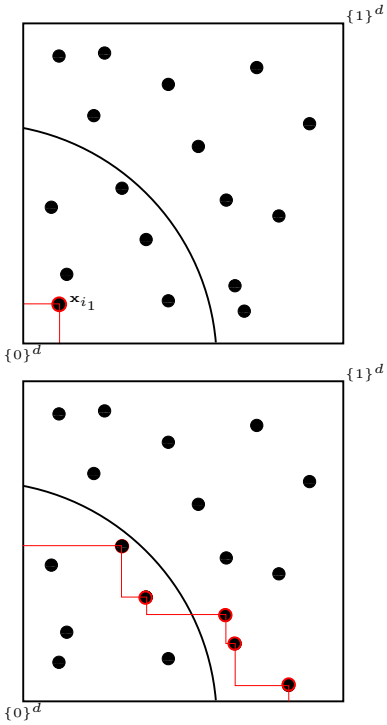
Figure 3: L'ensemble $\mathbb{W}(p)$ est représenté en gris pour différentes valeurs de p en dimension 2.

Cela signifie que q peut être borné par le minimum et le maximum des valeurs de g prisent sur E. Néanmoins, le minimum et le maximum de g ne sont pas très informatifs pour estimer un quantile. Pour affiner l'encadrement de q on propose de trouver un sous-ensemble de E tel que les hypothèses sur les volumes sont respectées.

Pour faire cela on part d'un sous-ensemble A de E que l'on va enrichir de points de E. Pour la borne supérieure (resp. inférieure), le point ajouté est celui qui contribue le moins à la mesure de $\mathbb{V}^-(A)$ (resp. $\mathbb{V}^+(A)$). Les algorithmes 5.1 et 5.2 décrivent cette méthode. On illustre sur la figure 4 les premières étapes de l'algorithme 5.2 pour $d = 2$.

Algorithme 5.1: Obtenir une borne inférieure pour q à partir d'un ensemble de points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

1. Étape d'initialisation: choisir $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\dots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^{d}(1-x^i)$
   Soit $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ et $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
2. Choisir le prochain point $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n))$
3. Soit $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ et $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
4. Si $vol \geq 1 - p$, répéter les étapes 2 et 3.
5. Retourne $\min_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.



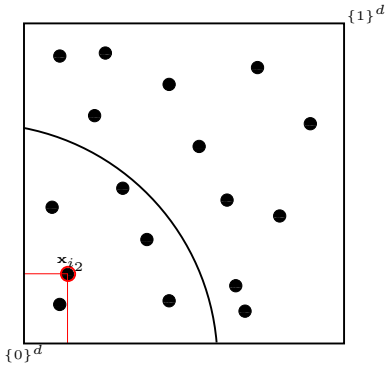


Figure 4: Illustration de l'algorithme 5.2 for $d = 2$. **Haut**: le point entouré $\mathbf{x}_{i_1}$ minimise la contribution du volume. **Bas**: on arrête l'algorithme lorsque $\mu(\mathbb{V}^-(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k})) \geq p$. Il est nécessaire de faire une évaluation à g seulement sur les points entourés.

Algorithme 5.2: Obtenir une borne supérieure pour q à partir d'un ensemble de points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

1. Étape d'initialisation: choisir $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\dots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^d x^i$
   Soit $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ et $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$
2. Choisir le prochain point $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n))$.
3. Soit $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ et $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$.
4. Si $vol \geq p$, répéter les étapes 2 et 3 .
5. Retourne $\max_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.

Un schéma d'échantillonnage séquentiel est maintenant possible. En effet, on sait mettre à jour les bornes du quantile et ainsi fournir un ensemble non-dominé. On propose de construire un estimateur $\widehat{F}$ de F et d'estimer q de la façon suivante.

$$\widehat{q} = \inf\{t \in \mathbb{R},\ \widehat{F}(t) \geq p\}. \tag{5.2}$$

Une fois l'étape d'initialisation réalisée, on construit une suite de vecteurs aléatoires uniformément distribués sur l'ensemble non-dominé. Dès que possible, on met à jour les bornes des quantiles, grâce aux algorithmes 5.1 et 5.2, ainsi que l'ensemble non-dominé. Ces étapes sont répétées jusqu'à ce que le budget d'appels au code soit épuisé. On note $(q_k^-)_{k\geq 1}$ et $(q_k^+)_{k\geq 1}$ les suites des bornes du quantile q ainsi obtenue.

En s'inspirant de l'estimation de probabilité sous contraintes de monotonie, un estimateur de F est donné par

$$\widehat{F}_n(t) = \frac{1}{n}\sum_{k=1}^{n} \left(p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{g(\mathbf{x}_k)\leq t\}}\right), \tag{5.10}$$

et est sans biais pour $t = q$. Le quantile q est alors estimé par

$$\widehat{q}_n = \inf\left\{t \in [q_n^-, q_n^+],\ \widehat{F}_n(t) \geq p\right\}. \tag{5.11}$$

Ce schéma d'échantillonnage assure la convergence des bornes (voir proposition 5.6). Par

construction, on en déduit que $\widehat{q}_n$ converge aussi vers q. Comme les conditions classique d'un théorème limite central ne sont pas pas vérifiées par $\widehat{F}_n(q)$ c'est également le cas pour $\widehat{q}_n$. Néanmoins, on peut obtenir des propriétés d'encadrement et de convergence pour $\widehat{F}_n$.

Proposition 5.7. Pour tout $t \in \mathbb{R}$, $F(\min(t,q)) \leq \mathbb{E}\left[\widehat{F}_n(t)\right] \leq F(\max(t,q))$ et

$$\mathbb{E}\left[\widehat{F}_n(t)\right] \underset{n\to+\infty}{\longrightarrow} F(q).$$

Conclusion

Dans cette thèse on s'est intéressé à l'adaptation au cas monotone des méthodes classiques d'estimation de probabilité et de quantile. Les méthodes séquentielles apparaissent plus adaptées car, à chaque nouvel appel au code les bornes obtenues sur les quantités d'intérêt sont mises à jour.

Plus spécifiquement, le comportement théorique des bornes ont été étudié. On a donné des conditions pour s'assurer que ces bornes convergent bien vers la probabilité ou le quantile visé. Ensuite, leur vitesse de convergence a été étudiée pour différents schémas de simulation. Comme attendu, une simulation séquentielle accélère significativement cette vitesse. Néanmoins, la monotonie apporte de moins en moins d'information lorsque la dimension augmente. Sous la condition que Γ soit convexe, on en a construit un estimateur adaptatif dont on contrôle la vitesse de convergence.

Concernant l'estimation d'une probabilité p, une nouvelle classe d'estimateur a été proposée. Mais une simulation séquentielle uniforme sur l'ensemble non-dominé semble être ce qu'il y a de plus pratique. On peut facilement obtenir un estimateur sans biais. La vitesse de convergence de la borne supérieure semble être équivalente pour cette simulation que pour la simulation optimale.

On a proposé une méthode d'encadrement d'un quantile à partir d'un ensemble de points. À partir de cet encadrement, on a pu construire un estimateur adaptatif de q.

Le calcul des bornes de la probabilité devient difficile lorsque la dimension augmente. De plus, il n'y a pas de raison pour que g reste globalement monotone et l'hypothèse de monotonie ne peut plus être exploitée.

Plusieurs thèmes n'ont pas été abordés dans cette thèse. Si un code n'est pas globalement monotone, on ne peut plus encadrer sûrement une probabilité ou un quantile. Mais des bornes conditionnelles peuvent probablement être utilisées pour guider l'estimation. Un autre aspect important en fiabilité est l'analyse de sensibilité. Cela consiste à quantifier l'influence des entrées sur une fonction de la sortie du code. Par exemple, sa variance, une probabilité de dépassement de seuil, un quantile... Il peut être intéressant de déterminer si la monotonie apporte de l'information sur ce type d'études. Certaines méthodes développées dans cette thèse ont été ou seront implémentées dans un package nommé MISTRAL de l'environnement logiciel R disponible sur le CRAN [98].

Extended abstract

A classical issue encountered by energy producers, like EDF, is to justify the reliability and safety of their production facilities. The physical phenomena involved in power plants production may carry damages to people, environment and goods if the structural reliability of these facilities is not insured. In France, studying the reliability of structures (or components) mainly relies on robustness testing: in *penalizing* running conditions, is the considered structure or component still be able to produce safely? For many highly reliable industrial components, feedback experience data are only related to safe situations, the situations considered as penalizing are never observed in practice from the commissioning and no failure has been observed. This is the case of numerous passive components of the French park exploited by EDF as, for instance, production vessels.

In absence of failure observations, structural reliability studies need to rely on mathematical modelling. One or several numerical models are designed and implemented by the specialists in order to simulate the considered phenomenon in usual and exceptional running conditions. The terms *computer code* or *computer model* will be extensively used in this document to refer to this implementation. Such codes often result from chaining of less complex codes, each of them being devoted to model one of the phenomena interacting on the supposed deterioration of the component.

A numerical model g, provided it is validated, allows to explore critical configurations, defined by the choice of an input parameter vector $\mathbf{x}$[4], and to determine if those configurations generate to a risk of failure. The latter can crudely be defined as follows: "the forcing (stress) $C(\mathbf{x})$[5] is equal to or upper than the resistance $R(\mathbf{x})$ of the structure (or component)". Therefore, defining generically $g(\mathbf{x}) = R(\mathbf{x}) - C(\mathbf{x})$, an input configuration will be considered as generating a failure if $g(\mathbf{x}) \leq 0$.

Beyond checking if several penalizing configurations do not generate failures, industrial safety studies currently use two complementary indicators :

- the probability of failure $\mathbb{P}(g(\mathbf{X}) \leq 0)$, by assuming that the input parameters $\mathbf{X}$, the knowledge of which being often uncertain, can randomly move in realistic ranges ;
- the *dual* indicator defined by the difference between resistance and stress Z_α such that $P(g(\mathbf{X}) \leq Z_\alpha)$ be lower than a given *acceptable threshold* α established by a safety rule.

Computing these two indicators allows in parallel to solve (at least partially) the inverse problem of determining the input configurations generating failures.

[4] that typically mix running parameters (e.g., pressure), environmental forcing (e.g., temperature) and, between others, features of materials.

[5] For instance an injection of cold water in a steel vessel brought at high temperature.

In practice, such complex codes are considered as "black boxes", which means that it is difficult or impossible to have a precise knowledge of all implemented functions and equations in due time. Therefore so-called *intrusive* exploration techniques, especially those founded on the differentiability of computer models, can not be carried out. Computer code exploration and computation of reliability indicators have to be conducted in practice using non-intrusive techniques, based on simulation.

Monte Carlo-type methods, that immediately appear natural when dealing with simulation, in theory allow to get statistical estimators of these indicators. Nonetheless, a huge simulation cost (in time or/and memory space) is often consubstantial to the complexity and precision of the computer model, that may forbid in practice the use of several of these methods. This is why a vast area of so-called *accelerated* (or variance reduction) techniques has known, these last years, a significant rise. The methodologies developed in this area aims to design *clever* numerical simulation experiments, allowing such estimations at weak computational cost. This thesis stands explicitly in this field of research.

Properly speaking, the "black box" interpretation of computer models is not fair in the reality of reliability studies. Indeed, the notation of penalizing configuration necessarily implies a global monotonicity of the phenomenon: the risk increases with any stress and decreases when any resistance increases. Therefore it appears reasonable to assume that a contribution which is less constraining that a sure configuration will itself be sure. In many cases, this monotonicity describes the behaviour of the phenomenon - and the computer code too, provided it is validated - with respect to its most influential variables. About safety studies the advantage of monotonicity is a priori considerable since it allows to surround the values of indicators by two bounds, in a deterministic sense rather than a probabilistic sense (using typically a confidence interval).

In this thesis work it is alternatively considered that the computer model can be interpreted as a "grey box", the monotonicity of which being known (with respect to uncertain inputs). Some parallel works are conducted at EDF R&D to check the completeness of this hypothesis, but they are integrated within this document. Another base hypothesis is that the probabilistic distributions of input parameters are known and independent. This assumption of independence can be relaxed under some conditions, which are studied in another specific research program at EDF R&D. Besides, very few additional hypotheses are done on the computer model: especially, it can simulate discontinuous physics (sometimes typical of critical configurations[6]) and present so-called *edge effects*. Finally, the computer model output may, in some cases, be reduced to a simple binary expression:

$$Z(\mathbf{x}) = \mathbb{1}_{\{g(\mathbf{x})\leq 0\}},$$

which allows to consider the code, in every sense, as a tool of decision (and not still the “simple" modelling of a physical phenomenon). In such a context, the notion of partial order on decisions appear as essential since it generates monotonicity, that impose a hierarchy between decisions. This notion of partial order will be at the heart of the works presented here.

Thus, this work deals with using monotonicity properties of computer codes for improving the estimation of reliability indicators, in a context where the reduction or a given limitation of the computational budget is required. This improvement is about the accuracy of bounds

[6]For instance, the physics of a river can be discontinuous close to flood configuration, which is characterized by a dike submersion.

previously evoked and the convergence properties of statistical estimators located between the bounds.

More precisely, three problems were studied during this Ph.D. thesis.

1. The theoretical behaviour of probability bounds, which are random objects since they are build from designs of simulated numerical experiments; building the designs ensuring a good convergence of bounds is a key aspect of this investigation; besides this study conducted us to build and study a *meta-model* of the limit state (failure) surface $\{\mathbf{x}, g(\mathbf{x}) = 0\}$.

2. The accelerated estimation of the failure probability defined above, especially by elaborating sequential designs of experiments.

3. The accelerated estimation of the dual quantile, using similar techniques.

Finally, a real case-study is deeply treated in a dedicated chapter, that deals with the reliability of some EDF production components. Their study constituted the industrial motivation of this thesis.

The main research results obtained during the thesis are more formally summarized in the following sections.

Adaptation of monotonicity constraints (Chapter 2)

In a first work the monotonic hypothesis are adapted to the most classical methods of probability estimation. The monotonic property is defined in the following definition.

Definition 2.1. Let $g : \mathbb{U} \subset \mathbb{R}^d \to \mathbb{R}$, g is said globally monotonic if g is monotonic relatively to each of its input.

Each input has either a unfavourable or favourable effect towards the reliability. To simplify the construction, without loss of generality it is considered that g is globally increasing. The underlying transformation is common to estimate a probability in engineering worl (see Sections 1.6 and 1.7).

The probability to estimate is defined by

$$p = \mathbb{P}(g(\mathbf{X}) \leq q),$$

with $\mathbf{X}$ a random vector with independent components. Transforming g allows to consider that $\mathbf{X}$ is a random vector uniformly distributed on $[0, 1]^d$ (see Section 2.2). It must be noticed that this transformation require to know the monotonicity of g according to each of its inputs. Finally, consider without loss of generality that $q = 0$.

The use of the monotonicity hypothesis is based on a partial order on $\mathbb{R}^d$.

Definition 2.2. Let $\mathbf{x} = (x_1, \dots, x_d), \mathbf{y} = (y_1, \dots, y_d) \in \mathbb{R}^d$ such that for all $i = 1, \dots, d,\ x_i \leq y_i$. The partial order of dominance is denoted $\mathbf{x} \preceq \mathbf{y}$. It is said that $\mathbf{y}$ dominates $\mathbf{x}$.

This means for all $\mathbf{x}, \mathbf{y} \in [0,1]^d$ such that $\mathbf{x} \preceq \mathbf{y}$ then $g(\mathbf{x}) \leq g(\mathbf{y})$. If it is known that $\mathbf{y}$ leads to an undesirable event then $\mathbf{x}$ leads also to the feared event. Such information is obtained without computing $g(\mathbf{x})$.

To simplify the presentation, denote

$$\begin{aligned}
\mathbb{U}^- &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) \leq 0\}, \\
\mathbb{U}^+ &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) > 0\}, \\
\Gamma &= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = 0\}.
\end{aligned}$$

These two sets represent the set of configurations leading respectively to a safety event, to an undesirable event, and situation located on the limit surface separating these two sets. Let A be a set of $[0,1]^d$. Denote

$$\begin{aligned}
\mathbb{U}^-(A) &= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^-} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \preceq \mathbf{x}\}, \\
\mathbb{U}^+(A) &= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^+} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \succeq \mathbf{x}\},
\end{aligned}$$

with $\mathbb{U}^-(\emptyset) = \{0\}^d = (0, \dots, 0)$ et $\mathbb{U}^+(\emptyset) = \{1\}^d = (1, \dots, 1)$.

The main interest of the monotonic hypothesis is summarised in the two following equations:

$$\begin{aligned}
\mathbb{U}^-(A) \subset \mathbb{U}^- \subset [0,1]^d \backslash \mathbb{U}^+(A), \\
\mu(\mathbb{U}^-(A)) \leq p \leq 1 - \mu(\mathbb{U}^+(A)),
\end{aligned}$$

with μ the Lebesgue measure on $\mathbb{R}^d$.

Then, Γ and the probability p can be bounded in a deterministic sense (surely). Since the number of runs of g is limited in practice, the set A is summarises in general to a set of points. The following notations are extensively used in this thesis. Let $(\mathbf{X}_k)_{k \geq 1}$ be a sequence of points or random vectors. Denote in the remainder of this work

$$\begin{aligned}
\mathbb{U}_n^- &= \mathbb{U}^-(\mathbf{X}_1, \dots, \mathbf{X}_n), \\
\mathbb{U}_n^+ &= \mathbb{U}^+(\mathbf{X}_1, \dots, \mathbf{X}_n), \\
\mathbb{U}_n &= [0,1]^d \backslash (\mathbb{U}_n^- \cup \mathbb{U}_n^+), \\
p_n^- &= \mu(\mathbb{U}_n^-), \\
p_n^+ &= 1 - \mu(\mathbb{U}_n^+).
\end{aligned}$$

Figure 5 illustrates this construction, given A a set of points. The Lebesgue measure of the sets represented in gray provides two bound for p.

From previous evaluations, the monotonic hypothesis allows to know the sign of g on some points. Such property can be easily exploited with the standard Monte Carlo method. Let $(\mathbf{X}_k)_{k \geq 1}$ be a sequence of independent random vectors uniformly distributed on $[0,1]^d$. Let n be the available total number of run, a Monte Carlo estimator of p is

$$\hat{p}_n^{MC} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{X}_i) \leq 0\}}. \tag{1.5}$$

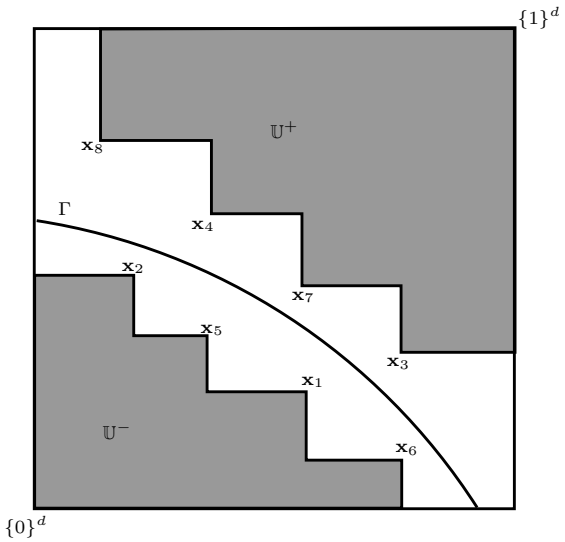


Figure 5: Illustration for $d = 2$ of $\mathbb{U}^-(A)$ and $\mathbb{U}^+(A)$ with $A = \{\mathbf{x}_1, \ldots, \mathbf{x}_8\}$.

Assume that $g(\mathbf{X}_1) > 0$. If $\mathbf{X}_2$ dominates $\mathbf{X}_1$ then without another run it is known that $g(\mathbf{X}_2) > 0$. The sign of g is known on two points while a unique evaluation by the numerical code has been made. Taking into account this construction, p can be estimated by

$$\frac{1}{N_n} \sum_{k=1}^{N_n} \mathbb{1}_{\{g(\mathbf{X}_k) \leq 0\}}, \tag{2.5}$$

where N_n is the number of simulations made with a budget of n runs to g. The number N_n is randomised. Indeed, at each step k the probability that evaluating $g(\mathbf{X}_k)$ is useless is equal to $1 - \mathbb{E}[p_{k-1}^+ - p_{k-1}^-]$. Nevertheless, the bounds of p are not exploited in the construction of this estimator.

Exploiting the monotonicity allows to increase, randomly, the total number of simulations. Nonetheless, the bounds of p are not totally used.

Once the bounds are obtained, the probability p can be rewritten as

$$p = p_n^- + (p_n^+ - p_n^-)\mathbb{P}(\mathbf{X} \in \mathbb{U}^- | \mathbf{X} \in \mathbb{U}_n).$$

This expression is close to the one obtained by *multivel simulation* described in Section 1.8. If these bounds are known, the probability p can be estimated by

$$\hat{p}_N = p_n^- + \frac{p_n^+ - p_n^-}{N} \sum_{k=1}^{N} \mathbb{1}_{\{\mathbf{X}_k^{(n)} \in \mathbb{U}^-\}}, \tag{2.7}$$

where $(\mathbf{X}_k^{(n)})_{k \geq 1}$ is a sequence of independent random vectors uniformly distributed on $\mathbb{U}_{n-1}$. This stratety is not optimal. The knowledge of the indicator function is not used to update the bounds of p.

Finally, it seems necessary to simulate, at each step n, on the *non-dominated* set

$$[0,1]^d \backslash (\mathbb{U}_{n-1}^- \cap \mathbb{U}_{n-1}^+).$$

In this way, the bounds and the non-dominated set are updated after each run of the numerical code.

Theoretical studies on the behaviour of the deterministic bounds (Chapitre 3)

The deterministic bounds of p can be used to build an efficient estimator of p. Nevertheless, it must be ensured that they converge toward p. Besides the convergence of these bounds, the convergence of the sequence of sets $(\mathbb{U}_k^-)_{k\geq 1}$ has been studied. The Hausdorff distance, defined below, is a good ingredient for this study.

Definition 3.2. Let $\|\cdot\|_q$ be the $\mathcal{L}^q$ norm on $\mathbb{R}^d$, that is for $0 < q < +\infty$, $\|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}$, and for $q = +\infty$, $\|\mathbf{x}\|_\infty = \max_{i=1,\dots,d} x_i$. Let (A, B) be two non-empty subsets of the normed vector space $([0,1]^d, \|\cdot\|_q)$. The Hausdorff distance $d_{H,q}$ is defined by

$$d_{H,q}(A,B) = \max(\sup_{\mathbf{y}\in A}\inf_{\mathbf{x}\in B}\|\mathbf{x}-\mathbf{y}\|_q;\ \sup_{\mathbf{x}\in B}\inf_{\mathbf{y}\in A}\|\mathbf{x}-\mathbf{y}\|_q).$$

A naive way to ensure the convergence is to simulate on the limit surface Γ. Obviously, this is impossible in practice but this idea provides a track to do it. Theoretical results state that it is needed to simulate infinitely often around Γ (see Proposition 3.2). The two studied frameworks of simulations are the standard Monte Carlo and the nested uniform sampling on the non-dominated space. Using these two strategies ensure the convergence of the deterministic bounds as well as the convergence of sets $\mathbb{U}_n^-$ and $\mathbb{U}_n^+$ respectively towards $\mathbb{U}^-$ and $\mathbb{U}^+$.

Then, the rate of convergence of $\mathbb{U}_n^-$ and $\mathbb{U}_n^+$ is examined. Under some regularity constraints (see Definition 3.3) this rate is known for the sequence $(\mathbb{U}_{n-1}^-)_{n\geq 1}$.

Proposition 3.4. Let $(\mathbf{X}_k)_{k\geq 1}$ be a sequence of iid random variables uniformly distributed on $[0,1]^d$ and $(\tilde{\mathbf{X}}_k)_{k\geq 1}$ be a sequence of independent and identically distributed random variables uniformly distributed on $\mathbb{U}^-$. Denote $\tilde{\mathbb{U}}_n^- = \mathbb{U}^-(\tilde{\mathbf{X}}_1,\dots,\tilde{\mathbf{X}}_n)$. Let $(F_n)_{n\geq 1}$ be a sequence of measurable subsets of $[0,1]^d$ such that for all $n \geq 1, \mathbb{U}_n^- \subset F_n \subset [0,1]^d \backslash \mathbb{U}_n^+$. Then

(1) $d_{H,2}(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{a.s.} 0$ and $\mu(F_n) \xrightarrow[n\to+\infty]{a.s.} p$.

(2) If $\mathbb{U}^-$ is regular, then almost surely

$$d_{H,2}(\tilde{\mathbb{U}}_n^-, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

(3) Furthermore, if $\mathbb{U}^+$ is also regular, and if g is continuous, then almost surely

$$d_{H,2}(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

For the rate of convergence of the bounds, let us start with $d = 1$ and a standard Monte Carlo framework. Proposition 3.5 states that

$$\begin{aligned} n(p - p_n^-) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1), \\ n(p_n^+ - p) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1), \\ \mathbb{E}[p_n^+ - p_n^-] &= \frac{2}{n+1} - \frac{1}{n+1}\left(p^{n+1} + (1-p)^{n+1}\right), \end{aligned}$$

where $\mathcal{E}xp(\lambda)$ is the exponential distribution with density $f_\lambda(x) = \lambda \exp(-\lambda x)\mathbb{1}_{\{x\geq 0\}}$.

Such distribution is not optimal because the knowledge of the deterministic bounds is not exploited. Moreover, it becomes more and more difficult to update the bounds while n increases. Indeed, the probability that X_n update the bounds is approximately equal to $1/n$. More and more simulations are required to update these bounds. To accelerate the rate of convergence, it would have been better to simulate at each step n on the interval $]p_{n-1}^-, p_{n-1}^+[$. Proposition 3.6 provides the mean distance between the bounds obtained from such framework

$$\frac{1}{2^n} \leq \mathbb{E}[p_n^+ - p_n^-] \leq \left(\frac{3}{4}\right)^n.$$

As expected, taking into account the information provided by the bounds accelerate significantly the rate of convergence. Study the case $d = 1$ is not relevant since it is equivalent to find the zero of an increasing function. It is then interesting to understand the influence of the dimension on these quantities. Being more general consider that $d \geq 2$. For $d = 1$ the limit surface Γ is unique and is equal to $\{p\}$. This is no more true for greater dimensions. It becomes difficult to obtain similar results for a given p. The study is restricted to $p = 1$. Proposition 3.7 provides an equivalent results than Proposition 3.6 for the sequence $(\mathbb{U}_k^-)_{k\geq 1}$. Indeed, for $d = 1$, $d_{H,q}(\mathbb{U}_n^-, [0,1]) = 1 - p_{n-1}^-$.

Proposition 3.7. Assume $\Gamma = \{1\}^d$. Let $(\mathbf{X}_k)_{k\geq 1}$ be a sequence of iid random vectors uniformly distributed on $[0,1]^d$. For $0 < q < +\infty$ denote $A(1,q) = 1$ and for $d \geq 2$,

$$A_{d,q} = \frac{1}{dq^{d-1}} \prod_{i=1}^{d-1} B(i/q, 1/q),$$

with $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$. For all $n \geq 1$, let $\mathbb{U}_n^- = \mathbb{U}^-(\mathbf{X}_1, \ldots, \mathbf{X}_n)$.

(1) If $0 < q < +\infty$ then

$$(A_{d,q}n)^{1/d} d_{H,q}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d).$$

(2) If $q = +\infty$ then

$$n^{1/d} d_{H,\infty}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d),$$

where $\mathcal{W}(1,d)$ is the Weibull distribution with scale parameter 1 and shape parameter d having cumulative density function $F(t) = 1 - e^{-t^d}$ for all $t \geq 0$.

An order of magnitude can besides be obtained on the mean distance of the bounds. Proposition 3.8 provides an order of magnitude of the mean distance between the bounds:

$$\mathbb{E}[1 - p_n^-] \underset{n\to+\infty}{\sim} \frac{\log(n)^{d-1}}{n(d-1)!}.$$

Exploiting the monotonicity becomes less interesting when the dimension increases. Moreover, in this context the numerical code may no longer be globally monotonic. The monotonic hypothesis is then more adapted for low dimensions. The use of a sequential framework of simulations seems essential to make the best use of the monotonic hypothesis. To estimate Γ, a sequential approach is more interesting. If Γ is convex (or concave) an adaptive estimator can be built from many

linear classifiers based on *Support Vector Machines* (SVM). Each linear classifier is calibrated from some design. The adaptive property comes from that for each new simulations, a new linear classifier is added to the current one. It is not necessary to rebuild the whole estimator at each step. When the computing time is an important factor, this sequential construction provides a real benefit. The construction of this estimator is now described.

Let $D_n = (\mathbf{X}_i, y_i)_{1\leq i\leq n} \in [0,1]^d \times \{-1,1\}$ be a design of experiments where $y_i = 1$ if $g(\mathbf{X}_i) > 0$ and -1 otherwise. Let $\Xi_n^+ = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^+$ and $\Xi_n^- = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^-$ and for all $\mathbf{x} \in \Xi_n^+$ define $h_{\mathbf{x}}$ a hyperplane separating $\mathbf{x}$ from Ξ_n^-. A classifier f_n is defined by

$$\begin{aligned} f_n : [0,1]^d &\to \{-1,+1\} \\ \mathbf{y} &\mapsto \begin{cases} -1 & \text{if for all } \mathbf{X} \in \Xi_n^+,\ h_{\mathbf{X}}(\mathbf{y}) \leq 0 \\ +1 & \text{otherwise,} \end{cases} \end{aligned}$$

and denote

$$F_n = \{\mathbf{x} \in [0,1]^d,\ f_n(\mathbf{x}) = -1\}$$

This construction is illustrated in dimension $d = 2$ on Figure 6. Theorem 3.1 provides the main properties of f_n and F_n.

Theorem 3.1. Assume $\mathbb{U}^-$ is convex, then

(1) f_n is globally increasing.

(2) For all $\mathbf{X} \in \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, $sign(g(\mathbf{X})) = f_n(\mathbf{X})$.

(3) The set F_n is a convex polyhedron.

(4) Furthermore if $(\mathbf{X}_k)_{k\geq 1}$ is a sequence of independent random vectors uniformly distributed on $[0,1]^d$, then

$$d_{H,2}(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{a.s.} 0,$$

and almost surely,

$$d_{H,2}(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

Probability estimation (Chapter 4)

The aim of this chapter is to provide an estimator of the probability p. This estimator must have better properties that the one provided in [16]. It must be unbiased, have a lower variance, reduce the upper bound of the searched probability and be easily tractable. Equation (4.6) provides a general expression of an unbiased estimator of p

$$\widetilde{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^- + \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^-\}} \right), \tag{4.6}$$

where $\omega_{1,n}, \ldots, \omega_{n,n}$ is a sequence of positive real values such that $\sum_{k=1}^n \omega_{k,n} = 1$. Estimator $\widetilde{p}_n$ is unbiased if for all $k \geq 1$ and for all $\mathbf{x} \in \mathbb{U}_{k-1} \cap \mathbb{U}^-$, $f_{k-1}(\mathbf{x}) > 0$. Classical results in importance sampling help to determine the sequence of probability density functions $(f_{k-1})_{k\geq 1}$ such that the variance of $\widetilde{p}_n$ becomes null:

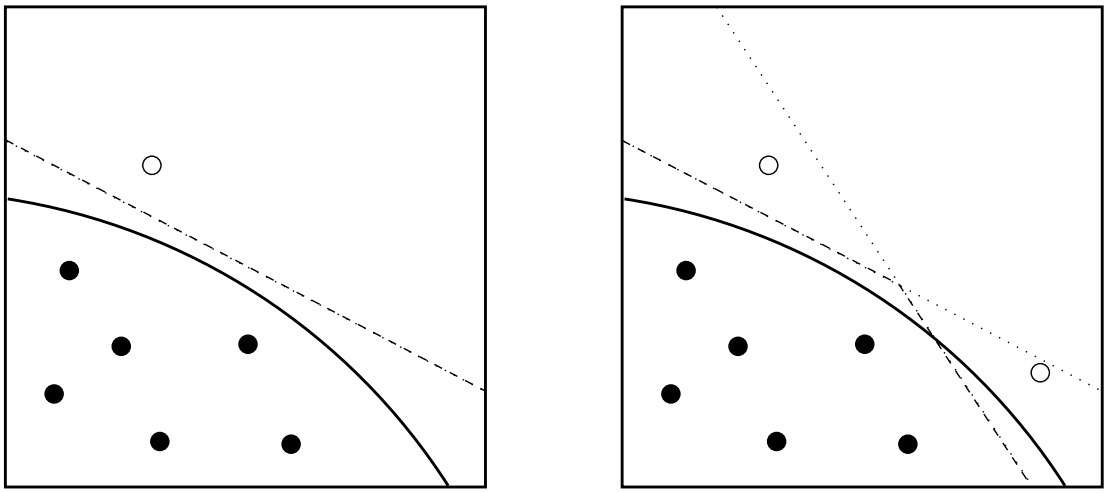


Figure 6: Construction of the classifier based on SVM. The plain line represent Γ and black (resp. white) points are in $\mathbb{U}^-$ (resp. $\mathbb{U}^+$). The dotted lines represents $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ for some $\mathbf{X}$ in $\mathbb{U}^+$. **Left**: there is one point in $\mathbb{U}^+$ then the dashed line is both $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ and the frontier of the classifier. **Right**: Dotted lines represent the two sets $\{\mathbf{x} \in [0,1]^d, h_{\mathbf{X}}(\mathbf{x}) = 0\}$ for $\mathbf{X}$ among the two white points. Dashed line represent the frontier of the classifier.

$$f_{k-1}(\mathbf{x}) = \frac{\mathbb{1}_{\{\mathbf{x} \in \mathbb{U}^- \cap \mathbb{U}_{k-1}\}}}{p - p_{k-1}^-}. \tag{4.7}$$

Obviously, this optimal density cannot be used in practice but gives ideas for an effective choice of importance distribution. It states that the simulation must be done uniformly on $\mathbb{U}^- \cap \mathbb{U}_{k-1}$. This implies that only the lower bound is updated. Since one of the objectives is to reduce as much as possible the upper bound, the following estimator can be built:

$$\widehat{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^+ - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}} \right). \tag{4.8}$$

Now, the variance of $\widehat{p}_n$ is minimum, if at each step k, the simulation is uniformly distributed on $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$. As previously, this cannot be done in practice. The aim is to simulate according to the optimal density while maintaining the unbiased condition. Consider at step k an estimator $\widehat{\mathbb{U}}_{k-1}^+$ of $\mathbb{U}^+ \cap \mathbb{U}_{k-1}$. If it is is sufficiently accurate, a uniform distribution appears to be close to the optimal distribution. Since it is impossible to check that $\widehat{\mathbb{U}}_{k-1}^+ \supset \mathbb{U}^+ \cap \mathbb{U}_{k-1}$, the estimator can be biased. To fix this issue, it is suggested to simulate $\mathbf{X}_k$ as follow

$$\mathbf{X}_{k-1} \sim \begin{cases} \mathcal{U}(\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}_{k-1}^+) & \text{with probability } \varepsilon_{k-1}, \\ \mathcal{U}(\widehat{\mathbb{U}}_{k-1}^+) & \text{with probability } 1 - \varepsilon_{k-1}, \end{cases} \tag{4.10}$$

The choice of ε_{k-1} is crucial to minimise the variance of the estimator. But this choice implies to known $\mathbb{U}^+$. Since $\mathbb{U}^+$ must be estimated at each step the use of such methods is very time-consuming in practice. Finally, a simple case is studied. Let $(\mathbf{X}_k)_{k \geq 1}$ a sequence of random vectors such that for all $k \geq 1, \mathbf{X}_k$ is uniformly distributed on $\mathbb{U}_{k-1}$. Denote

$$\bar{p}_k = p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-) \mathbb{1}_{\{g(\mathbf{X}_k) \leq 0\}},$$

and an unbiased estimator of p becomes

$$\widehat{p}_n = \frac{1}{n} \sum_{k=1}^{n} \bar{p}_k.$$

It must be noticed that the uniform distribution on the non-dominated space is not so far to the optimal one. Indeed, it is equivalent to choose $\widehat{\mathbb{U}}_{k-1}^{+} = \mathbb{U}^{+} \cap \mathbb{U}_{k-1}$. Numerical results obtained on an example show that $\mathbb{E}[1-\varepsilon_{k-1}] = \mathbb{E}[(p_{k-1}^{+} - p)/(p_{k-1}^{+} - p_{k-1}^{-})]$ is close to 1. This means with high probability the simulations are in $\mathbb{U}^{+} \cap \mathbb{U}_{k-1}$.

The convergence of the upper bound has been compared for two frameworks of simulations. The first one is the uniform distribution on the non-dominated set and the second one is the optimal density. As said previously, this density is not usable in practice. The use of a reject method allows to simulate uniformly on $\mathbb{U}^{+}$. Numerical experiments have shown that the use of the optimal density do not reduce significantly the upper bound compared to a uniform simulation on the non-dominated set.

Finally, the uniform distribution seems to be more tractable in practice and provides easily an unbiased estimator. The optimal framework does not seem to be more efficient to reduce the upper bound. Moreover, it does not require to build at each step the estimator $\mathbb{U}^{+} \cap \mathbb{U}_{k-1}$. These remarks reveal that the considered estimator has good properties and is easily usable in practice.

Nevertheless, it remains difficult to control it. Indeed, the theoretical tools state that under verifiable condition the fluctuations of the estimator are too small to obtain a central limit theorem.

Quantile estimation (Chapitre 5)

The aim of this chapter is to provide a consistent estimator as well as two bounds for a quantile. To bound a probability, it is sufficient to know the sign of g on a set of points. This approach can no longer be conducted is no more available for quantile estimation. Let F be the cumulative distribution function of $g(\mathbf{X})$. The p-quantile of $g(\mathbf{X})$ is defined by

$$q = \inf\{t \in \mathbb{R},\ F(t) \geq p\}. \tag{5.1}$$

Some definitions are now provided to simplify the presentation.

Definition 5.1. Let $A \subset [0,1]^d$. Define

$$\mathbb{V}^{-}(A) = \bigcup_{\mathbf{x} \in A} \{\mathbf{u} \in [0,1]^d :\ \mathbf{u} \preceq \mathbf{x}\},$$
$$\mathbb{V}^{+}(A) = \bigcup_{\mathbf{x} \in A} \{\mathbf{u} \in [0,1]^d :\ \mathbf{u} \succeq \mathbf{x}\}.$$

Definition 5.2. Let $\alpha \in]0,1[$ and S be a set in $[0,1]^d$. The set S is said α-monotonic if for all $\mathbf{u}, \mathbf{v} \in S$, $\mathbf{u}$ is not strictly dominated by $\mathbf{v}$ and if $\mu(\mathbb{V}^{-}(S)) = \alpha$.

Obtaining the bounds for a quantile is entirely based on the following proposition.

Proposition 5.3. For all $\alpha \in]0,1[$ assume that Γ_α is an α-monotonic set. Let $(p^{-}, p^{+}) \in [0,p[\times[p,1[$, then

$$\min_{\mathbf{x} \in \Gamma_{p^-}} g(\mathbf{x}) \leq q \leq \max_{\mathbf{x} \in \Gamma_{p^+}} g(\mathbf{x}).$$

It must be noticed that $\Gamma = \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\}$ is p-monotonic. In practice, it is difficult to build such sets. It can require a too high number of runs to find the minimum and the

maximum of g on this set. Two technical constraints are to be taken into account. The first one is to build easily a monotonic set. The second one is that a limited runs of g is required to find its minimum and its maximum. The solution that we propose is the construction of a monotonic set built from a set of points. Let $\mathbf{x} = (x^1, \dots, x^d)$ be a point of $[0,1]^d$. The boundary of $\mathbb{V}^-(\mathbf{x})$ is a monotonic set. As g is globally increasing, its maximum on the boundary of $\mathbb{V}^-(\mathbf{x})$ is reached in $\mathbf{x}$. The boundary of $\mathbb{V}^-(\mathbf{x})$ is p-monotonic if $x^1 \cdots x^d \geq p$. If so, it comes that $g(\mathbf{x}) \geq q$. A similar construction allows to obtain a lower bound for q. These constructions are summarised in Proposition 5.4.

Proposition 5.4. Let $p \in]0,1[$ and $d \geq 2$. Denote

$$\mathbb{W}^-(p) = \left\{ \mathbf{u} = (u^1, \dots, u^d) \in [0,1]^d, \prod_{i=1}^{d} (1 - u^i) \geq 1 - p \right\},$$
$$\mathbb{W}^+(p) = \left\{ \mathbf{u} = (u^1, \dots, u^d) \in [0,1]^d, \prod_{i=1}^{d} u^i \geq p \right\}.$$

For all $(\mathbf{u}, \mathbf{v}) \in \mathbb{W}^-(p) \times \mathbb{W}^+(p)$ it comes

$$g(\mathbf{u}) \leq q \leq g(\mathbf{v}).$$

Denoting $\mathbb{W}(p) = [0,1]^d \backslash (\mathbb{W}^-(p) \cup \mathbb{W}^+(p))$, then

$$\mu(\mathbb{W}(p)) = (1-p) \sum_{k=0}^{d-1} \left[\frac{(-\log(1-p))^k}{k!} \right] + p \sum_{k=0}^{d-1} \left[\frac{(-\log(p))^k}{k!} \right] - 1.$$

The set $\mathbb{W}(p)$ represents the set of points in $[0,1]^d$ where the sign of $g(.) - q$ is unknown. This construction does not depend on g. Let g be a globally increasing function, then $\{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\} \subset \mathbb{W}(p)$ where q is the p-quantile of $g(\mathbf{X})$.

This initialisation step is particularly interesting when p tends to 0 or 1. Indeed, $\mu(\mathbb{W}(p))$ tend to 0 while p tends to 0 or 1. Figure 7 provides the set $\mathbb{W}(p)$ for different values of p.

Since an initialisation step is done, the aim is to find two bounds for q from a set of points E. Proposition 5.3 states that

$$\mu(\mathbb{V}^-(E)) > p \Rightarrow \max_{\mathbf{x} \in E} g(\mathbf{x}) \geq q,$$
$$\mu(\mathbb{V}^+(E)) > 1 - p \Rightarrow \min_{\mathbf{x} \in E} g(\mathbf{x}) \leq q.$$

This means that q can be bounded by the minimum and the maximum of g on E. Nevertheless, the minimum and the maximum of g are not very informative to estimate a quantile. To refine these bounds, it is proposed to build a subset A of E such that $\mu(\mathbb{U}^-(A)) \geq p$ or $\mu(\mathbb{U}^+(A)) \geq 1 - p$. To do this, let us start with a subset A of E which is then enriched with some points of E. For the upper bound (resp. lower), the added point is the one that bringing the least contribution to the measure of $\mathbb{V}^-(A)$ (resp. $\mathbb{V}^+(A)$). Algorithms 5.1 and 5.2 describe this method. It is illustrated in Figure 4 the first steps of Algorithm 5.2 for $d = 2$.

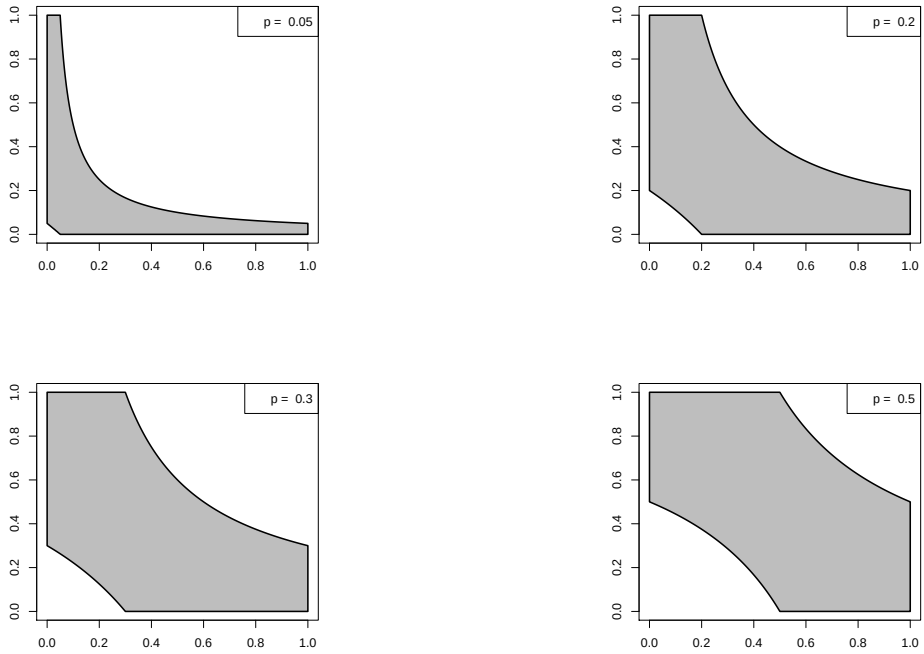


Figure 7: The set $\mathbb{W}(p)$ is represented in gray for different values of p in dimension 2.

Algorithm 5.1: getting a greater bound for q from a set of points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

1. Initialisation step : choose $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\ldots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^d (1-x^i)$
   Set $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ and $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
2. Choose the next point as $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n))$
3. Set $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ and $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
4. If $vol \geq 1 - p$, repeat steps 2 and 3.
5. Return $\min_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.

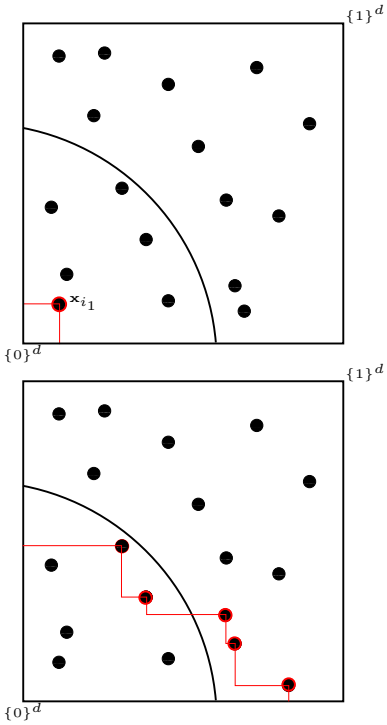
Algorithm 5.2: getting a lower bound for q from a set of points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$

1. Initialisation step : choose $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\ldots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^d x^i$
   Set $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ and $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$
2. Choose the next point as $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n))$.
3. Set $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ and $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$.
4. It $vol \geq p$, repeat steps 2 and 3.
5. Return $\max_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.

A sequential framework of simulations is now possible. Indeed, the bounds can be updated and provide a non-dominated set. Let $\widehat{F}$ be an estimator of F. It is proposed to estimate q as follows:

$$\widehat{q} = \inf\{t \in \mathbb{R},\ \widehat{F}(t) \geq p\}, \tag{5.2}$$

Once the initialisation step is done, a sequence of random vectors uniformly distributed on



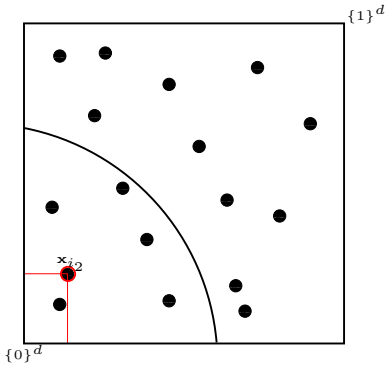


Figure 8: Illustration of Algorithm 5.2 for $d = 2$. **Up**: the encircled point $\mathbf{x}_{i_1}$ minimises the contribution of the volume. **Down**: the algorithm is stopped when $\mu(\mathbb{V}^-(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k})) \geq p$. It is required to compute the value of g only on the encircled points.

the non-dominated set is produced. As soon as the bounds of q can be updated using Algorithms 5.1 or 5.2, the non-dominated set can be also updated. These steps are repeated until the budget of evaluations by the numerical code is reached. Let $(q_k^-)_{k\geq 1}$ and $(q_k^+)_{k\geq 1}$ be the sequences of bounds of the quantile q obtained.

Following probability estimation under monotonicity constraints, an estimator of F is given by

$$\widehat{F}_n(t) = \frac{1}{n}\sum_{k=1}^{n}\left(p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{g(\mathbf{x}_k)\leq t\}}\right). \tag{5.10}$$

and is unbiased for $t = q$. The quantile q is then estimated by

$$\widehat{q}_n = \inf\left\{t \in [q_n^-, q_n^+],\ \widehat{F}_n(t) \geq p\right\}. \tag{5.11}$$

This sequential framework of simulations ensures that the bounds converge to q (see Proposition 5.6). From construction, it is deducted that $\widehat{q}_n$ converge towards q. As a central limit theorem is not available for $\widehat{F}_n(q)$ with verifiable conditions, the same result applies to $\widehat{q}_n$. Nevertheless, some consistency properties can be obtained for $\widehat{F}_n$.

Proposition 5.7. For all $t \in \mathbb{R}$, $F(\min(t, q)) \leq \mathbb{E}\left[\widehat{F}_n(t)\right] \leq F(\max(t, q))$ and

$$\mathbb{E}\left[\widehat{F}_n(t)\right] \underset{n\to+\infty}{\longrightarrow} F(q).$$

Conclusion

In this thesis, the adaptation of classical methods for probability and quantile estimations under monotonicity constraints has been examined. Sequential methods appear clearly to be more adapted since each new run of the numerical code allow to update the deterministic bounds.

The behaviour of these bounds has been studied. The conditions to ensure the convergence towards the probability and the quantile have been provided. Especially, their rate of convergence have been studied for different simulations framework. As expected, a sequential framework accelerates significantly their rate of convergence. Nonetheless, the monotonic property provides less and less information while the dimension increases. Under the constraint that Γ is convex, an adaptive estimate of Γ has been proposed and its convergence can be controlled.

A new construction to estimate a probability has been provided. Sequential uniform sampling on the non-dominated set seems to be the more tractable method in practice. A unbiased estimator can be easily deducted.

Alternatively, a method have been provided to bound a quantile from a set of points. From these bounds, an adaptive estimator of q has been built.

The computation of the bounds of p becomes difficult while the dimension increases. Moreover, while the dimension increases there is no reason that g is still globally monotone. The monotonic hypothesis becomes more and more difficult to use when the dimension increases.

Many themes have not been studied in this thesis. If a numerical code is not globally monotonic, a bound for a probability or a quantile are no longer available. But conditional bounds can be probably used to guide the estimation. Another important aspect in reliability is sensitivity analysis. This consists in quantifying the influence of the input on a function of the output of the numerical code. For example, its variance, a probability, a quantile... It can be useful to determine if the monotonic hypothesis bring information in such studies. Some of the methods provided in this thesis have or will be implemented in a package so-called MISTRAL coded in R and available on CRAN [98].

Introduction

Context

As many energy producers, EDF must justify that its productions plants are safe in running or stopping condition. This safety justification can be defended by studying the risk engendered from the occurrence of an undesirable event and can be defined as the product of a cost with the probability that such event occur. For example, a replacement and/or manufacturing cost, damages to people and environment. Assessing such probabilities is a key thematic in structural reliability and constitutes the primary topic of this thesis. More generally, the aim is to provide reliability indicator of an industrial component.

When the studied component is highly reliable, an undesirable event may have never happened and then never be observed. The reliability experiments are to costly (e.g. destructive testing) or dangerous. A possible solution is to implement, under the form of a so-called *computer code*, a numerical model g that simulates the considered phenomenon. In the industrial context explored in this thesis it depends on different types of physics as thermodynamics or mechanics. It is difficult to known the whole of calculus involved in this model. The numerical code is then considered as *black-box*: only the input and the output of the numerical can be known. Moreover, assume that the code is deterministic: a given input provides a unique output

Once this numerical code built, it is necessary to characterise the input of this code. These inputs represent physical configurations which the component is submitted. One configuration is defined by the intrinsic physical characteristics of the component as its shape, the materials used in its construction and the properties of its environments, temperature, pressure, mechanics constraints... Nonetheless, they can be split in two classes. The first one contains the parameters which varying during the life of the component and the second one contains the known and fixed parameters. For each of these configurations, the numerical model provides a reliability index. To summarise, when a physical configuration $\mathbf{X}$ is given as input to the numerical code, it says that if the configuration leads to an undesirable event or no.

The searched probability is the proportion of these configurations leading to an undesirable event against all possible configurations. This ratio does not take into account of the frequency of occurrence of configurations and it is assumed that they have the same influence on the reliability. To represent these differences a weight can be associated with each of these configurations. More a configuration occurs often, more the associated weight is high. These weights can be determined from observations or by the knowledge of the physical properties of the component. Choosing normalised weights, summing to one, these configurations are represented by a random vector. This does not mean that $\mathbf{X}$ is random, but it takes into account that the behaviour of component is not uniform with all parameters. In the case where the inputs of the numerical code are not represented by a random vector, it is equivalent to take identical weights. Even if g is a black-box, the knowledge of the studied physical phenomenon allows to make hypotheses on

the numerical code. For example, if a configuration leads to a safe event, it is reasonable to say that a less restrictive configuration leads also to a safe event. Conversely, if a configuration leads to an undesirable event, a more severe configuration for the component leads in all probability to an undesirable event. In this thesis it is assumed that these properties are verified by the numerical code. It is said that the code is monotone.

Most of probability estimation methods allow to control, with a given confidence level, the produced estimator. This confidence level is usually fixed by the user at 95%. Obtain a confidence level at 100% is equivalent to say that the probability is comprise between 0 and 1, which is not informative. One of the main advantages of the monotonicity hypothesis is that a non-trivial confidence interval at 100% for the searched probability can be obtained. Another advantage is that the reliability of a configuration can be known without any more evaluations by g. In practice, this is equivalent to decide if a configuration leads or not to an undesirable event only from evaluations already made by the numerical code.

Mathematical context

Being more formal and general, consider a numerical code g :

$$\begin{aligned} g : \mathbb{R}^d \times \mathbb{R}^m &\to \mathbb{R} \\ (\mathbf{x}, \mathbf{m}) &\mapsto g(\mathbf{x}, \mathbf{m}). \end{aligned}$$

It is now considered only the input $\mathbf{x}$ and the univariate output $g(\mathbf{x})$. A configuration (or input) $\mathbf{x}$ generates an undesirable event if $g(\mathbf{x}) \leq q$ with q fixed by the user. In this thesis, the probability that such an event occurs is examined first. To do this, the input vector is represented by a random vector $\mathbf{X}$ and then $g(\mathbf{X})$ is a random variable. Assume moreover that the probability density function $f_{\mathbf{X}}$ of $\mathbf{X}$ is known. The probability p that an undesirable event occur is then defined by

$$p = \mathbb{P}(g(\mathbf{X}) \leq q) = \mathbb{E}[\mathbb{1}_{\{g(\mathbf{X}) \leq q\}}] = \int_{\mathbb{R}^d} \mathbb{1}_{\{g(\mathbf{x}) \leq q\}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

Conversely, quantile estimation aim to estimate the threshold q when p is known. Denote F the cumulative distribution of $\mathbf{X}$ defined as follow

$$F(t) = \mathbb{P}(g(\mathbf{X}) \leq t).$$

To illustrate this, assume that F is continuous and strictly increasing, then

$$F(q) = p.$$

Probability and quantile estimation are connected since if a quantity is known (p or q), the other one must be estimated.

Even if g is considered black-box, the knowledge of the studied physical phenomenon allows to make some hypotheses on the numerical code. In this thesis it is considered that if an input $\mathbf{x}$ leads to the undesirable event then a more severe configuration leads also to the undesirable event.

For example, consider $\mathbf{X} = (R, S)$ where the two random variables R and S represent respectively a resistance and some constraints. Assume that an undesirable event occur if $g(R, S) = R - S \leq 0$. The undesirable event $\{R - S \leq 0\}$ has a lower probability to occur if the resistance R increases and/or if the solicitation S decreases. Being more general, let $g : \mathbb{R}^2 \to \mathbb{R}$

Input	Distribution	parameters	Physical representation
X^1	truncated Weibull	(1.8, 0.00309 , 0.00005, 0.05)	depth of a flaw
X^2	Log-normal	(-1.516, 0.504)	Ratio height/length
X^3	Normal	(0,1)	Resistance
X^4	Normal	(0,1)	Constraints

Table 1: Input of the industrial numerical code.

be a function and let (R, S) be a random vector such that if R increases (resp. S decreases) then g increases. Let r, s two points such that $g(r, s) \geq 0$, then for all $\varepsilon_R, \varepsilon_S \geq 0$ it comes

$$g(r + \varepsilon_R, s - \varepsilon_S) \geq 0,$$
$$\mathbb{P}(R \geq r, S \leq s) \geq p.$$

These two equations summarise the whole advantage of monotonicity hypothesis. From a design of experiments, without any more evaluation by the numerical code, the sign of g can be known on some points of the input space. Moreover, the probability p can be bounded with a confidence level at 100%. Indeed, the upper bound does not depend on g. From now, these bounds are so-called the deterministic bounds of p.

Industrial case study

This thesis has been motivated by the following case study. The reliability of some nuclear components of a pressurized water vessel must be proved. This thesis focuses on a component considered to be not replaceable.

To do this, many physical constraints are taking into account to build a model. The neutrons which initiate the nuclear reaction may collide with the component. It can loses matter when such collisions occurs and then becomes less resistant. The high temperature of its environment could cause the deformation of the studied component. Moreover, manufacturing defect must be taking into account.

The studied model represents the propagation of a flaw on such a component. In such situations, this flaw may spread in the component. It must be justify that this eventual spreading does not involve the loss of integrity of the component.

Among all parameters taking as input, only some of them can be represented by random variables. The numerical code is monotonic according to some of them. As this thesis focuses on globally monotonic functions, only these inputs are considered. The other ones are set to their nominal values. The distribution of the input random vector is summarised in Table 1. The position of the flaw and general resistance and constraints of the component are taking into account.

Objectives

Three problems have been studied in this thesis.

The theoretical behaviour of probability bounds, which are random objects since they are build from designs of simulated numerical experiments. Building the designs ensuring a good convergence of bounds is a key aspect of this investigation. Besides this study conducted us to build and study a *meta-model* of the limit state (failure) surface.

The second problem is the estimation of the probability p. Recall that the probability is defined by

$$p = \mathbb{P}(g(\mathbf{X}) \leq q).$$

This threshold q represents a reliability constraint which may come from physical or regulatory constraints. Once this threshold is fixed, an estimator of this probability allows to justify the reliability of the studied component. As evoked above, the monotonicity hypothesis allows to obtain two bounds sure at 100% for p. If the upper bound is lower that an acceptable probability fixed by the user then the reliability is completely proved. For example, assume that the frequency of the rise in the water level knowing the height of a dike must be studied. The undesirable event occur if during one year the height of the water lever is greater than the height of the dike. An upper bound for the probability is then an upper bound for the frequency of overflow in one year. Conversely, the lower bound can indicate if the frequency of overflow is too high, then confirm the height of the dike is to low.

The third studied problem during this thesis is quantile estimation. This means that the probability is fixed and the associated threshold q must be estimated. As for probability estimation, the monotonic hypothesis allows to bound with a confidence level at 100% the searched quantile. This hypothesis allows to determine if a configuration leads to the feared event or not without additional runs to the numerical computer code. This type of bounding permit to determine if a component verify the fixed reliability constraints. In practice, the bounds enable to adjust the design of the component during its conception step. By taking the previous example, assume that the aim is to prevent of a one-hundred-year flood. In this case, the height of the dike corresponds to a p-quantile with $p = 10^{-2}$. An upper bound for q ensure a sufficient height for the dike with a confidence level at 100%. If the height of the current dike is lower than the lower bound of the quantile then a one-hundred-year flood will occur with probability one.

Outline

In Chapter 1, classical methods for probability estimation are presented. It is split in four mains sections. The first one examines Monte Carlo methods. Descriptions of the standard Monte Carlo, importance sampling techniques and Quasi-Monte Carlo methods are provided. The second one focuses on engineering methods used in structural reliability. The estimation of a probability is obtained by solving an optimisation problem. The third one is based on using sequential designs. This means that each simulation is conducted in function of the knowledge of the previous one. Lastly, method based one surrogate (meta-modelling) are examined in the situation where the numerical is time-consuming. Such technique can provides an estimation of g which is not time-consuming.

In the two firsts sections of Chapter 2, existing and adapted classical methods of probability estimation are presented, that take into account of the monotonicity of g.

Information provided by the deterministic bounds around a probability is examined in Chapter 3. The rate of convergence as well as the convergence in law of these quantities are studied. Different designs are considered. The first one is a Monte Carlo based design and the second is a sequential framework that exploits the knowledge of deterministic bounds. Moreover, a sequential estimator of the limit state $\{\mathbf{x}, g(\mathbf{x}) = 0\}$ verifying monotonic constraint is built.

The aim of Chapter 4 is to provide a probability estimator which have better properties than the existing one. A general form of such estimator is provided and its construction is discussed.

In Chapter 5, the construction of probability estimation is adapted to quantile estimation. An initialisation step is provided then a method to get deterministic bounds of a quantile is built. Finally it is presented an estimator of this quantile using the probability estimator provided in the fourth chapter.

The methods provided in Chapter 4 and 5 are tested on the industrial case in Chapter 6.

Chapter 1

State of the art: non-intrusive estimation of an exceedance probability

Résumé Dans ce premier chapitre, un état de l'art des méthodes non-intrusives d'estimation d'espérance, et plus particulièrement de probabilité en sortie d'un code numérique, est présenté. Ce chapitre débute par la méthode de Monte Carlo et inroduit des méthodes de réduction de variance. Une méthode issue du monde ingénieur dédiée à l'estimation de probabilité est ensuite présentée. Cette méthode transforme le problème d'estimation en un problème d'optimisation. Ensuite, des méthodes qui utilisent séquentiellement l'information des simulations sont introduites. Enfin, lorsque le nombre d'évaluations par le code numérique est limité, celui-ci peut être remplacé par un *métamodèle* de coût de calcul négligeable, les méthodes tirant parti de cette substitution sont examinées.

Abstract In this first chapter, a state of the art of non-intrusive methods for integral estimation, and more precisely to estimate a probability, is provided. This chapter first examines Monte Carlo methods then variance reduction methods. A method coming from engineering studies and specifically tuned for probability estimation is then provided. This method consists in transforming the estimation problem into an optimisation problem. Next, sequential methods which use pieces of information provided by simulations are introduced. Lastly, when the feasible number of evaluations by the numerical code is limited, it can be replaced by a *meta-model* with a negligible cost, methods that take benefit are examined.

1.1 Introduction

Denote by $(\Omega, \mathcal{A}, \mathbb{P})$ a probability space. Let $\mathbf{X}$ be a d-dimensional random vector on $\mathbb{U} \subset \mathbb{R}^d$ with a known probability density function $f_{\mathbf{X}}$. Denote by $g : \mathbb{U} \subset \mathbb{R}^d \to \mathbb{R}$ a measurable function and define

$$p = \mathbb{P}(g(\mathbf{X}) \leq 0) = \mathbb{E}[\mathbb{1}_{\{g(\mathbf{X})\leq 0\}}] = \int_{\mathbb{U}} \mathbb{1}_{\{g(\mathbf{x})\leq 0\}} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}. \tag{1.1}$$

It must be noticed that in practice the threshold is not necessary 0. Without loss of generality, if the threshold is equal to a real value q then $g(.)$ is transformed in $g(.) - q$. In a structural reliability context, such numerical code are complex and each run of g can be time-consuming

(e.g. one second/minute/hour/day... per run). In practice, this means that the number of evaluations by g is limited. Moreover, the complexity of the model implies that g is considered black-box: for a given input $\mathbf{x}$, only the value $g(\mathbf{x})$ is known. The methods described in this section are non-intrusive.

The aim of this chapter is to make an overview of many methods to estimate the probability (1.1). Some of them require regularity assumptions on g (for example differentiability). It is split in four main parts. In the first one, the most common methods based on Monte Carlo-type sampling are examined. The main disadvantage is while p decreases, less and less simulations fall within the set of interest $\{\mathbf{x} \in \mathbb{U},\ g(\mathbf{x}) \leq 0\}$. Thus, for a given precision the size of sample increases when p becomes small. To address this problem, *importance sampling* techniques [63] are introduced to increase the probability of simulating in $\{\mathbf{x} \in \mathbb{U},\ g(\mathbf{x}) \leq 0\}$. Using an *importance density*, importance sampling methods modify the definition of p. Moreover, in a given sense, the ideal importance distribution can be theoretically found, but is unreachable in practice. Therefore a vast area of these techniques incorporate approaches to build importance distribution close to the optimal one.

Other methods refer to engineering practice. This class of methods, known under the name FORM/SORM (First/Second Order Reliability Methods), transforms the estimation of p in solving an optimisation problem. Besides an estimation of p, this method provides an estimation of the *limit surface* $\{\mathbf{x} \in \mathbb{U},\ g(\mathbf{x}) = 0\}$. as well as the *design point*. This point is the most probable input configuration leading to the undesirable event $\{\mathbf{x} \in \mathbb{U},\ g(\mathbf{x}) \leq 0\}$.

Monte Carlo methods do not use dynamically the simulations to improve the estimation of p or the limit state surface. In a third part, sequential methods, which aim to do so, are examined. *Splitting* method describes p as a product of greater probabilities theoretically easier to estimate. The last section provides general method using meta-model: a simplified model (non time-consuming) which mimics g. Then, replacing g by such meta-model, all previous methods can be used once again.

1.2 Standard Monte Carlo

1.2.1 General description

The standard Monte Carlo [103] is based on the strong law of large numbers (SLLN). Let $r > 0$ and Z be a random variable, one said $Z \in \mathbb{L}^r$ if $\mathbb{E}[|Z|^r] < +\infty$. Let $Z \in \mathbb{L}^1$ be a real random variable, the SLLN states that if $(Z_n)_{n\geq 1}$ is a sequence of random variables independent and identically distributed (iid) with the same law as Z, then

$$\lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} Z_i \overset{a.s.}{=} \mathbb{E}[Z]. \tag{1.2}$$

An estimator of $I = \mathbb{E}[Z]$ is given by

$$\hat{I}_n^{MC} = \frac{1}{n}\sum_{i=1}^{n} Z_i. \tag{1.3}$$

From linearity of the expectation $\mathbb{E}[\hat{I}_n^{MC}] = I$ and if $Z \in \mathbb{L}^2$, the variance of $\hat{I}_n^{MC}$ is equal to $\mathrm{Var}(\hat{I}_n^{MC}) = \sigma^2/n$ with $\sigma^2 = \mathrm{Var}(Z)$. An empirical way to get more information is to study a sample of estimators of I. In practice, it can be impracticable to do this if the number of

simulations Z_i is limited. The central limit theorem (CLT) is commonly used to control the fluctuations of such estimators. If $Z \in \mathbb{L}^2$, the central limit theorem states that

$$\frac{\sqrt{n}}{\sigma}\left(\hat{I}_n^{MC} - I\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0,1),$$

where $\xrightarrow[n\to+\infty]{\mathcal{L}}$ denotes the convergence in distribution while n goes to infinity. The notation $\mathcal{N}(0,1)$ represents the standard Gaussian distribution with probability density function $\phi(x) = e^{-x^2}/\sqrt{2\pi}$ and cumulative distribution function $\Phi(x) = \int_{-\infty}^{x} \phi(t)dt$. For a sufficiently large n the expectation I is, with probability $(1-\alpha) \in [0,1]$, in the following confidence interval

$$CI_\alpha = \left[\hat{I}_n^{MC} - \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{n}}, \hat{I}_n^{MC} + \Phi^{-1}(1-\alpha/2)\frac{\sigma}{\sqrt{n}}\right], \tag{1.4}$$

where Φ^{-1} denote the inverse of the function Φ. In practice, the variance σ^2 of Z can be unknown and the confidence interval (1.4) cannot be used. Nonetheless, an unbiased estimator of σ^2 is given by

$$\widehat{\sigma}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(Z_i - \hat{I}_n^{MC}\right)^2,$$

and from Slutsky's theorem it is deduced that

$$\frac{\sqrt{n}}{\widehat{\sigma}_n}\left(\hat{I}_n^{MC} - I\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}(0,1).$$

Finally, replacing the estimation $\widehat{\sigma}_n^2$ by σ^2 in (1.4) provides the following $100(1-\alpha)\%$ confidence interval

$$CI_\alpha = \left[\hat{I}_n^{MC} - \Phi^{-1}(1-\alpha/2)\frac{\widehat{\sigma}_n}{\sqrt{n}}, \hat{I}_n^{MC} + \Phi^{-1}(1-\alpha/2)\frac{\widehat{\sigma}_n}{\sqrt{n}}\right].$$

Being more general, the central limit theorem can be extended in multidimensional case. Let $(\mathbf{Z}_n)_{n\geq 1}$ be a sequence of iid random vectors with mean-vector I and covariance matrix Σ. Denote

$$\hat{I}_n^{MC} = \frac{1}{n}\sum_{k=1}^{n}\mathbf{Z}_k.$$

The multidimensional central limit theorem states that

$$\sqrt{n}\left(\hat{I}_n^{MC} - I\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}_d(\mathbf{0}_d, \Sigma),$$

where $\mathcal{N}_d(\mathbf{0}_d, \Sigma)$ represents the multivariate Gaussian distribution with d-dimension mean-vector $\mathbf{0}_d = (0,\ldots,0) \in \mathbb{R}^d$ and covariance matrix Σ.

1.2.2 Application to probability estimation

Recall that $\mathbf{X}$ is a d-dimensional random vector on $\mathbb{U} \subset \mathbb{R}^d$ with probability density function $f_{\mathbf{X}}$ and denote $p = \mathbb{P}(g(\mathbf{X}) \leq 0) = \mathbb{E}[\mathbb{1}_{\{g(\mathbf{X})\leq 0\}}]$. Let $(\mathbf{X}_n)_{n\geq 1}$ be an iid sequence of random vectors distributed as $\mathbf{X}$. From Equation (1.2), the probability p can be estimated by

$$\hat{p}_n^{MC} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\{g(\mathbf{X}_i)\leq 0\}} \xrightarrow[n\to+\infty]{p.s.} \mathbb{P}[g(\mathbf{X}) \leq 0], \tag{1.5}$$

with a variance equals to

$$Var\left[\hat{p}_n^{MC}\right] = \frac{p(1-p)}{n} \leq \frac{1}{4n}, \tag{1.6}$$

which depends on the unknown probability p. As for the general case, the variance of $\mathbb{1}_{\{g(\mathbf{X})\leq 0\}}$ can be estimated by

$$\widehat{\sigma}_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(\mathbb{1}_{\{g(\mathbf{X}_i)\leq 0\}} - \hat{p}_n^{MC}\right)^2,$$

and a confidence interval at $100(1-\alpha)\%$ becomes

$$CI_\alpha = \left[\hat{p}_n^{MC} - \Phi^{-1}(1-\alpha/2)\frac{\widehat{\sigma}_n}{\sqrt{n}}, \hat{p}_n^{MC} + \Phi^{-1}(1-\alpha/2)\frac{\widehat{\sigma}_n}{\sqrt{n}}\right]. \tag{1.7}$$

The standard Monte Carlo is one of the simplest method to use in practice. The precision of the estimation does not depends on the dimension and it is sufficient to retain the value of g on the sample. Moreover, no hypotheses have to be made on the regularity of the function g. It can be discontinuous, have no derivative and it is sufficient to know if the code is lower or greater than a given threshold. Nonetheless, the principal drawback of this method is the rate of convergence in $1/n$. For all $\varepsilon > 0$, the Chebyshev's inequality states that

$$\mathbb{P}(|\hat{p}_n^{MC} - p| > \varepsilon) \leq \frac{\sigma^2}{\varepsilon^2 n}.$$

The size n of the sample can be chosen to control the coefficient of variation defined by

$$CV_{MC} = \frac{\sqrt{Var\left[\hat{p}_n^{MC}\right]}}{\mathbb{E}(\hat{p}_n^{MC})} = \sqrt{\frac{1-p}{np}}.$$

Even if the coefficient of variation does not depends on g, it is not robust according to p:

$$CV_{MC} \underset{p\to 0}{\longrightarrow} +\infty.$$

A possible stopping criterion is to simulate until $CV_{MC} \leq 0.1$:

$$[CV_{MC} \leq 0.1] \Rightarrow \left[n \geq 10^2(1-p)/p\right]. \tag{1.8}$$

For a given precision, the size of the sample increases while p decreases. If the order of magnitude of p is 10^{-k}, Equation (1.8) states that n must be approximately greater than 10^{k+2}. When the computer code is time-consuming and p is small ($< 10^{-3}$) this method is not feasible in practice. The following sections summarise some variance reduction techniques for integral estimation or exclusively probability estimation.

1.3 Space filling methods

As said in the previous section, the standard Monte Carlo is inefficient to simulate rare events in practice. If there is no sample in one of the informative areas, the estimation of an expectation will have a large variance. This can be explained by the non-regularity of the sample. Since

(a) A standard Monte Carlo sample on $[0,1]^2$.

(b) A regular grid in dimension 2.

Figure 1.1: Representation of two different sampling techniques.

each elements of the sample is built independently, some empty areas can appear (see Figure 1.1).

In this section, several space filling methods are provided. The set of points obtained is used to compute integrals. This class of method are called Quasi-Monte Carlo. First, stratified sampling [29] is presented. This class of method consists in estimating the expectation separately in different subsets of the domain of integration. Second, the Latin Hypercube Sampling (LHS) developed in [83], a design method which can be compared with stratified sampling, is studied. If a sample of size n is required then n quartile of the initial distribution will be represented by the sample. Third, the low discrepancy sequences (see [91]) provide deterministic samples which are close to the theoretical uniform distribution in some sense. Lastly, maximin and minimax designs are presented and discussed.

1.3.1 Stratified Sampling

Stratified Sampling [29, 103] consists in splitting up the input space in strata and to estimate the expectation $I = \mathbb{E}[H(\mathbf{X})]$ knowing to be in one of the strata. In other words, the expectation I is rewritten as

$$I = \sum_{i=1}^{m} \mathbb{E}[H(\mathbf{X})|\mathbf{X} \in S_i]\mathbb{P}(\mathbf{X} \in S_i),$$

where $\cup_{i=1}^{m} S_i = \mathbb{R}^d$ with $S_i \cap S_j = \emptyset$ for $i \neq j$. If every $\rho_i = \mathbb{P}(\mathbf{X} \in S_i)$ are known, it remains to estimate $I_i = \mathbb{E}[H(\mathbf{X})|\mathbf{X} \in S_i]$. For example, it could be estimated by

$$\hat{I}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} H(\mathbf{X}_i),$$

where $(\mathbf{X}_j^i)_{j\geq 1}$ is a sequence of iid random vectors distributed as $\mathbf{X}^i \sim \mathcal{L}(\mathbf{X}|\mathbf{X} \in S_i)$ and $\sum_{i=1}^{m} n_i = n$. Finally, I is estimated by the following unbiased estimator

$$\hat{I}_n^{Strat} = \sum_{i=1}^{m} \rho_i \hat{I}_i,$$

the variance of which being

$$\mathrm{Var}(\hat{I}_n^{Strat}) = \sum_{i=1}^{m} \frac{\rho_i^2 \sigma_i^2}{n_i}, \tag{1.9}$$

with $\sigma_i^2 = \text{Var}(\mathbb{E}[H(\mathbf{X})|\mathbf{X} \in S_i])$. Setting $n_i = \rho_i n$ is an intuitive choice. Nonetheless, the variance in (1.9) is minimum with

$$n_i = n \frac{\rho_i \sigma_i}{\sum_{j=1}^{m} \rho_j \sigma_j},$$

and becomes

$$\text{Var}(\hat{I}_n^{Strat}) = \frac{1}{n} \left(\sum_{i=1}^{m} \rho_i \sigma_i \right)^2.$$

Without any more information on H, if $\mathbf{X}$ is a random vector of $\mathbb{R}^d$, each S_i can be one of the 2^d orthant[1] of $\mathbb{R}^d$.

1.3.2 Latin hypercube Sampling

The Latin Hypercube Sampling was developed by [83] in any dimension. It consists in splitting the input space in different strata and to simulate in some of them. The aim is to represent each marginal of $\mathbf{X}$ by a sample. To do this, the support of the probability density function of each marginal distribution is split in n subsets, representing the number of simulations available. Assume that $\mathbf{X} = (X^1, \dots, X^d)$ takes its values in $\mathbb{U} = S_1 \times \cdots \times S_d$. Then, each S_i is split in n subsets $S_{i,1}, \dots, S_{i,d}$ with same probability according to f_i, the probability density function of X^i. Thus, $S_i = \cup_{k=1}^n S_{i,k}$ with $\mathbb{P}(X_i \in S_{i,k}) = 1/n$ for all $k = 1, \dots, n$.

Let $\mathbf{X}_1, \dots, \mathbf{X}_n$ be n points in $\mathbb{U}$ with $\mathbf{X}_j = (X_j^1, \dots, X_j^d)$ and $(X_1^1, \dots, X_n^1)$ distributed on $S_{1,1} \times \cdots \times S_{1,n}$. Let $\pi_2, \dots, \pi_d$ be $d-1$ random and independent permutations of $\{1, \dots, n\}$. For all $k = 2, \dots, d$, denote $\pi_k(1, \dots, n) = (i_{k,1}, \dots, i_{k,n})$. Finally, for all $j = 1, \dots, n$, $\mathbf{X}_j$ is simulated on $S_{1,j} \times S_{2,i_{2,j}} \times \cdots \times S_{d,i_{d,j}}$.

Figure 1.2 illustrates this construction for $S_1 \times S_2 = [0,1]^2$. In Figures 1.2a and 1.2b the permutations π_2 are respectively equal to $\{4,2,1,5,3\}$ and $\{4,5,2,1,3\}$.

One of the main advantages of LHS is that the projection of a sample of size n in each of the d directions provides a new sample of size n. For example, in Figure 1.1b the projection of the sample provides 11 different points.

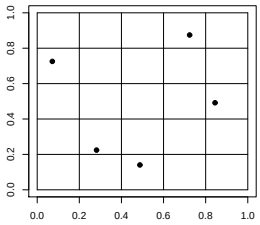
1.3.3 Low discrepancy sequences

The aim of space filling methods is to approximate the theoretical uniform distribution. If many choices of sample are available it may be appropriate to compare them. To do this, the discrepancy compares the empirical distribution of a sample with the theoretical uniform distribution. Without loss of generality consider that the aim is to simulate uniformly on $[0,1]^d$. Let $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a set of points in $[0,1]^d$. A definition of the discrepancy of $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ provided in [91], is denoted

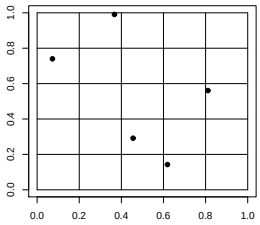
$$D(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sup_{S \in \mathcal{S}} \left| \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{\{\mathbf{x}_k \in S\}} - \mu(S) \right|,$$

where μ represents the Lebesgue measure in $\mathbb{R}^d$ and $\mathcal{S}$ is the sets of d-dimensional hyper-rectangle of the form $\prod_{i=1}^d [a_i, b_i[$ with $0 \leq a_i < b_i \leq 1$ for all $i = 1, \dots, d$. Since the supremum is computed

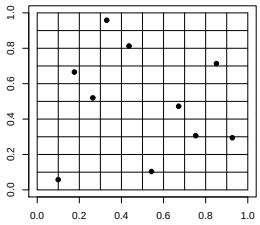
[1] An orthant in $\mathbb{R}^d$ is defined by $\{\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d, \varepsilon_i x_i \geq 0\}$ where $\varepsilon_i \in \{-1, 1\}$.



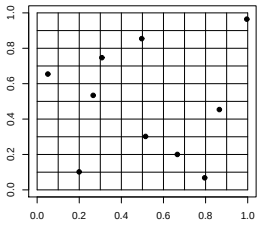
(a) LHS on $[0,1]^2$ with $n = 5$



(b) LHS on $[0,1]^2$ with $n = 5$



(c) LHS on $[0,1]^2$ with $n = 10$



(d) LHS on $[0,1]^2$ with $n = 10$

Figure 1.2: Representation of different LHS in dimension $d = 2$.

on all hyper-rectangles in $[0,1]^d$, the comparison between many sequences becomes infeasible in practice. To facilitate the computation, the so-called *star-discrepancy* [92] is a modified version of the discrepancy. It is defined as follow

$$D^*(\mathbf{x}_1, \dots, \mathbf{x}_n) = \sup_{S \in \mathcal{S}^*} \left| \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}_{\{\mathbf{x}_k \in S\}} - \mu(S) \right|,$$

where $\mathcal{S}^*$ is the sets of d-dimensional hyper-rectangle of the form $\prod_{i=1}^{d} [0, b_i[$ with $0 < b_i \leq 1$ for all $i = 1, \dots, d$. There is a relation between these two quantities

$$D^*(\mathbf{x}_1, \dots, \mathbf{x}_n) \leq D(\mathbf{x}_1, \dots, \mathbf{x}_n) \leq 2^d D^*(\mathbf{x}_1, \dots, \mathbf{x}_n).$$

In [92], the discrepancy and star-discrepancy of a finite set of points in dimension 1 are given. The values of D and D^* are difficult to obtain in higher dimension. Moreover, these bounds are not helpful to compare the discrepancy between two sets of points. Let E_1 and E_2 be two sets of points with same number of elements. The relation $D^*(E_1) \leq D^*(E_2)$ does not imply that $D(E_1) \leq D(E_2)$.

A sequence with a low discrepancy is called a *low-discrepancy sequence*. In practice, the use of such sequences is useful to compute integrals. Let $h : [0,1]^d \to \mathbb{R}$ be a measurable function and $(\mathbf{X}_k)_{k \geq 1}$ be a sequence of independent random vectors uniformly distributed on $[0,1]^d$. The strong law of large number states that

$$\lim_{n \to +\infty} \frac{1}{n} \sum_{i=1}^{n} h(\mathbf{X}_i) \overset{a.s.}{=} \mathbb{E}[h(\mathbf{X})] = \int_{[0,1]^d} h(\mathbf{x}) d\mathbf{x}.$$

For an equivalent error of approximation, a low-discrepancy sequence can require less evaluations by h than the uniform sample $\mathbf{X}_1, \dots, \mathbf{X}_n$. Let $(\mathbf{x}_k)_{k \geq 1}$ be a sequence of points in $[0,1]^d$ and h

be a function of bounded variation. In [67], it is shown that for all $n \geq 1$

$$\left| \frac{1}{n} \sum_{k=1}^{n} h(\mathbf{x}_k) - \int_{[0,1]^d} h(\mathbf{x}) d\mathbf{x} \right| \leq V(h) D^*(\mathbf{x}_1, \ldots, \mathbf{x}_n), \tag{1.10}$$

where $V(h)$ is the total variation of h in Hardy and Krause sense. In (1.10), the discrepancy and $V(h)$ have independent role. But in practice, the error strongly depends on both h and the sample. This upper bound is used to prove that the low-discrepancy sequence provides a consistent estimator of the integral. In the next paragraph some low-discrepancy sequences are examined.

Some of them are particular case of the so-called (t, m, d)-*net sequences* (see [91]). First, it is required to determine a positive integer b which takes the role of a basis. Then, a number of strata is determined as well as the number of points simulated on each of these strata. In basis b, the size of the sample is equal to b^m and the Lebesgue measure of each strata is equal to b^{t-m} and contains b^t points. An upper bound of the star-discrepancy of a (t, m, d)-net can be found in [91]:

$$\begin{aligned} &\frac{2^t}{n} \sum_{k=0}^{d-1} \binom{m-t}{k} \text{ for } b = 2, \\ &\frac{b^t}{n} \sum_{k=0}^{d-1} \binom{d-1}{k} \binom{m-t}{k} \left\lfloor \frac{b}{2} \right\rfloor^k \text{ for } b \geq 3. \end{aligned}$$

Several well known low-discrepancy sequences are provided in the following paragraphs (see also [79]).

Van der Corput sequence. The Van der Corput sequence provides a low-discrepancy sequence in dimension one which depends on a basis b. The n first terms $x_0, \ldots, x_{n-1}$ of the sequence is built as follow. For all $i = 0, \ldots, n-1$, the number i is decomposed in basis b. The number 0 is coded by 0 for any basis. If $i \geq 1$, it can be coded in basis b with $m(i, b)$ digits. This number of digits is the smaller integer such that $i \leq b^{m(i,b)}$. Thus,

$$m(i, b) = \begin{cases} 1 & \text{if } i = 0, \\ 1 + \left\lfloor \frac{\log(i)}{\log(b)} \right\rfloor & \text{if } i \geq 1, \end{cases} \tag{1.11}$$

and it comes

$$i = \sum_{k=0}^{m(i,b)-1} \alpha_k(i, b) b^k. \tag{1.12}$$

Finally, the n first terms of the Van der Corput sequence are defined by $\{x_0, \ldots, x_{n-1}\}$ with

$$x_i = \sum_{k=0}^{m(i,b)-1} \alpha_k(i, b) b^{-(k+1)}.$$

For example, the nine first terms of the Van der Corput sequence in basis $b = 3$ are

$$\begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 2 & 0 \\ 2 & 1 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 3^{-2} \\ 3^{-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 1/3 \\ 2/3 \\ 1/9 \\ 4/9 \\ 7/9 \\ 2/9 \\ 5/9 \\ 8/9 \end{pmatrix}.$$

Halton sequence The Halton sequence is the multidimensional version of the Van der Corput Sequence. Instead to simulate a sequence of Van der Corput of size dn, the Halton sequence is built with d Van der Corput sequences with d different basis. For all $i = 0, \dots, n-1$, the number i is decomposed in d basis $b_1, \dots, b_d$ where the kth number is denoted $\alpha_k(i, b_j)$. The n first terms of the Halton sequence are defined by $\{\mathbf{x}_0, \dots, \mathbf{x}_{n-1}\}$ with

$$\mathbf{x}_i = \left(\sum_{k=0}^{m(i,b_j)-1} \alpha_k(i, b_j) b_j^{-(k+1)} \right)_{j=1,\dots,d} \in [0,1]^d,$$

where $m(i, b_j)$ is constructed as in (1.11). In practice, the basis $b_1, \dots, b_d$ must be co-prime integers[2]. On Figure 1.3 are displayed four Halton sequences in dimension $d = 2$ for different couples of basis. Figures 1.3b and 1.3c highlight that the basis must be co-primes integer. Otherwise, some undesirable patterns can appear.

Hammersley sequence Hammersley sequence is a combination of a Halton sequence and a term depending on the size of the sample. In other words, using the same notations as in the previous paragraph, the Hammersley sequence is defined for all $i = 0, \dots, n-1$ by

$$\mathbf{x}_i = \left(\frac{i}{n}, \sum_{k=0}^{m(i,b_2)-1} \alpha_k(i, b_2) b_2^{-(k+1)}, \dots, \sum_{k=0}^{m(i,b_d)-1} \alpha_k(i, b_d) b_d^{-(k+1)} \right).$$

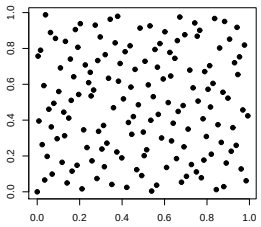
Sobol sequence Sobol sequence was developed in [109]. Such sequences can be built in any dimension. But to simplify the construction let us start with $d = 1$. At the difference with the other sequences presented above, the basis is equal to $b = 2$ for any dimension. Let $(x_0, \dots, x_{n-1})$ be a sequence of points in $[0, 1]$ such that for all $i = 0, \dots, n-1$

$$x_i = \sum_{k=1}^{m} t_k 2^{-k},$$

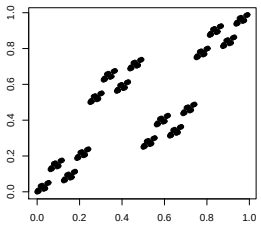
where $t_k \in \{0, 1\}$ for all $k \geq 1$. The construction of a Sobol sequence requires the representation in basis 2 of x_i which is denoted

$$(x_i)_2 = 0.t_1 t_2 \cdots t_m.$$

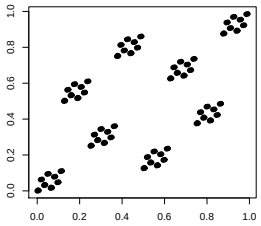
[2]Two integers are co-prime if the only positive integer which divides them is 1.



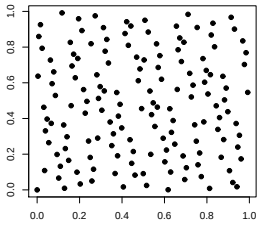
(a) A Halton sequence with $(b_1, b_2) = (2, 3)$.



(b) A Halton sequence with $(b_1, b_2) = (2, 4)$.



(c) A Halton sequence with $(b_1, b_2) = (2, 8)$.



(d) A Halton sequence with $(b_1, b_2) = (2, 11)$.

Figure 1.3: Representation of several Halton sequences in dimension $d = 2$.

For example $3/8 = 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$ and then $(3/8)_2 = 0.011$. The construction of such sequence requires a primal polynomial P in $\mathbb{Z}_2 = \{0, 1\}$:

$$P(z) = z^s + a_1 z^{s-1} + a_2 z^{s-2} + \cdots a_{s-1} z + 1,$$

with $a_1, \ldots, a_{s-1} \in \{0, 1\}$. Moreover, let $(m_k)_{k \geq 1}$ be a sequence of integers such that for $1 \leq k \leq s, m_k$ is an odd integer chosen arbitrary between 1 and 2^k (e.g. $2k + 1$) and for $k > s$,

$$(m_k)_2 = (2a_1 m_{k-1})_2 \oplus \cdots \oplus (2^{s-1} a_{s-1} m_{k-s+1})_2 \oplus (2^s m_{k-s})_2 \oplus (m_{k-s})_2, \tag{1.13}$$

where $\oplus$ represents the exclusive-or operator. For example $01001 \oplus 11011 = 10010$. Let $(\alpha_k(i, 2))_{k \geq 0}$ be the parameters of the decomposition in basis 2 of i as described in (1.12), the representation in basis 2 of x_i is

$$(x_i)_2 = \oplus_{k \geq 1} \alpha_k(i, 2)(v_k)_2,$$

with

$$v_k = m_k / 2^k.$$

Building a Sobol sequence in dimension $d \geq 2$ requires to build d Sobol sequences in dimension 1 with d different primal polynomial.

Let us present an example in dimension 1. Let $s = 3$ and $P(z) = z^3 + z^2 + 1$ then $a_1 = 1, a_2 = 0$ and let $m_1 = 1, m_2 = 3$ and $m_3 = 5$. From (1.13), it comes

$$\begin{aligned}
(m_4)_2 &= (2a_1 m_3)_2 \oplus (2^2 a_2 m_2)_2 \oplus (2^3 m_1)_2 \oplus (m_1)_2, \\
&= (10)_2 \oplus (0)_2 \oplus (8)_2 \oplus (1)_2, \\
&= 1010 \oplus 0000 \oplus 1000 \oplus 0001, \\
&= 0011.
\end{aligned}$$

Since $v_1 = 1/2, v_2 = 3/4, v_3 = 5/8$ and $v_4 = 3/16$, it comes

$$\begin{aligned}(v_1)_2 &= 0.1000,\\ (v_2)_2 &= 0.1100,\\ (v_3)_2 &= 0.1010,\\ (v_4)_2 &= 0.0011.\end{aligned}$$

Finally,

$$\begin{aligned}(x_0)_2 &= (0)_2 = 0,\\ (x_1)_2 &= (v_1)_2 = 0.1,\\ (x_2)_2 &= (v_2)_2, = 0.11,\\ (x_3)_2 &= (v_1)_2 \oplus (v_2)_2 = 0.01,\\ (x_4)_2 &= (v_3)_2 = 0.101,\\ (x_5)_2 &= (v_1)_2 \oplus (v_3)_2 = 0.001,\\ (x_6)_2 &= (v_2)_2 \oplus (v_3)_2 = 0.011,\\ (x_7)_2 &= (v_1)_2 \oplus (v_2)_2 \oplus (v_3)_2 = 0.111,\\ (x_8)_2 &= (v_4)_2 = 0.0011.\end{aligned}$$

Concluding on this example, the nine first points of this Sobol sequence is

$$(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8) = (0, 1/2, 3/4, 1/4, 5/8, 1/8, 3/8, 7/8, 3/16).$$

1.3.4 Optimised design

As seen in Figure 1.1a, uniform sampling clusters points and create empty areas. Several criteria of design was developed to fight against this phenomenon.

Maximin and minimax designs

Maximin and minimax designs, provided in [71], are based on two intuitive ideas. The aim is to find a set of points which minimises a criterion.

The maximin-distance criterion consists in maximising the minimal distance between the points of a sample. Let $\bar{\mathbf{x}}$ be a set of points in $[0,1]^d$, this criterion is equal to

$$mindist(\bar{\mathbf{x}}) = \min_{\substack{\mathbf{x},\mathbf{y}\in\bar{\mathbf{x}}\\ \mathbf{x}\neq\mathbf{y}}} \|\mathbf{x}-\mathbf{y}\|, \tag{1.14}$$

and the aim is to find $\bar{\mathbf{x}}$ such that (1.14) is maximum. This criterion means that the minimum distance of two points of a sample is maximum.

Minimax design provides a sample such that for each point of $[0,1]^d$, the distance with its nearest neighbour of the sample is minimal. This criterion is defined by

$$MinMax(\bar{\mathbf{x}}) = \max_{\mathbf{y}\in[0,1]^d} \min_{\mathbf{x}\in\bar{\mathbf{x}}} \|\mathbf{y}-\mathbf{x}\|. \tag{1.15}$$

Low discrepancy LHS

A method to minimise the discrepancy of a design of experiments based on LHS is provided by [70]. Many methods to optimise LHS are detailed in [34]. This problem can be seen as minimising a criterion by a set of points. If the criterion has not a closed form, simulated annealing can be used in practice. A LHS is chosen randomly among all LHS and it is accepted or rejected as the current minimiser of the criterion with a given probability. This step is repeated until reaching a stopping criterion.

1.4 Importance sampling

1.4.1 General description

The standard Monte Carlo does not ensure to obtain a sample in a domain which brings enough information to estimate accurately an expectation. For example, if the expectation is a small probability defined by $p = \mathbb{E}[\mathbb{1}_{\{g(\mathbf{X})\leq 0\}}]$, the event $\{g(\mathbf{X}) \leq 0\}$ must be simulated as frequently as possible. For this purpose, a random vector is introduced to modify the initial distribution $\mathbf{X}$ with the aim of concentrating the simulations in an informative area of the input space. Methods based on this mechanism are called *importance sampling* techniques [63, 103]. More generally, let $\mathbf{X}$ be a random vector on $\mathbb{R}^d$ with probability density function $f_{\mathbf{X}}$ and $H : \mathbb{R}^d \to \mathbb{R}$ such that $H(\mathbf{X}) \in \mathbb{L}^1$. Under mild assumptions, $\mathbb{E}[H(\mathbf{X})]$ can be rewritten as

$$\begin{aligned}\mathbb{E}[H(\mathbf{X})] &= \int_{\mathbb{R}^d} H(\mathbf{x}) \frac{f_{\mathbf{X}}(\mathbf{x})}{f_{\mathbf{Y}}(\mathbf{x})} f_{\mathbf{Y}}(\mathbf{x}) d\mathbf{x}, \\ &= \mathbb{E}\left[H(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{f_{\mathbf{Y}}(\mathbf{Y})}\right],\end{aligned}$$

with $\mathbf{Y}$ a random vector with probability density function $f_{\mathbf{Y}}$.

Definition 1.1 *Let $h : \mathbb{R}^d \to \mathbb{R}$, the support of h is defined by $supp(h) = \{\mathbf{x} \in \mathbb{R}^d,\ h(\mathbf{x}) \neq 0\}$.*

Equalities above hold if the support of $f_{\mathbf{Y}}$ contains the support of $f_{\mathbf{X}}$. Let $(\mathbf{Y}_n)_{n\geq 1}$ be an iid sequence of random vectors with the same law than $\mathbf{Y}$. If $H(\mathbf{Y}) f_{\mathbf{X}}(\mathbf{Y})/f_{\mathbf{Y}}(\mathbf{Y}) \in \mathbb{L}^1$, the strong law of large numbers states that

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{i=1}^{n} H(\mathbf{Y}_i) \frac{f_{\mathbf{X}}(\mathbf{Y}_i)}{f_{\mathbf{Y}}(\mathbf{Y}_i)} \overset{a.s.}{=} \mathbb{E}\left[H(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{f_{\mathbf{Y}}(\mathbf{Y})}\right]. \tag{1.16}$$

An estimator of $I = \mathbb{E}[H(\mathbf{X})]$ is then given by

$$\hat{I}_n^{IS} = \frac{1}{n} \sum_{i=1}^{n} H(\mathbf{Y}_i) \frac{f_{\mathbf{X}}(\mathbf{Y}_i)}{f_{\mathbf{Y}}(\mathbf{Y}_i)}. \tag{1.17}$$

Remark 1.1 *Estimator (1.17) is unbiased if $supp(f_{\mathbf{X}}) \subset supp(f_{\mathbf{Y}})$.*

If $H(\mathbf{Y}) f_{\mathbf{X}}(\mathbf{Y})/f_{\mathbf{Y}}(\mathbf{Y}) \in \mathbb{L}^2$, the variance of $\hat{I}_n^{IS}$ is equal to

$$\begin{aligned}\mathrm{Var}\left(\hat{I}_n^{IS}\right) &= \frac{1}{n} Var\left(H(\mathbf{Y}) \frac{f_{\mathbf{X}}(\mathbf{Y})}{f_{\mathbf{Y}}(\mathbf{Y})}\right), \\ &= \frac{1}{n}\left[\int \left(H^2(\mathbf{y}) \frac{f_{\mathbf{X}}^2(\mathbf{y})}{f_{\mathbf{Y}}(\mathbf{y})}\right) d\mathbf{y} - \mathbb{E}[H(\mathbf{X})]^2\right],\end{aligned} \tag{1.18}$$

and the variance of $H(\mathbf{Y})f_{\mathbf{X}}(\mathbf{Y})/f_{\mathbf{Y}}(\mathbf{Y})$ can be estimated by

$$(\widehat{\sigma}_n^{IS})^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(H^2(\mathbf{Y}_i)\frac{f_{\mathbf{X}}^2(\mathbf{Y}_i)}{f_{\mathbf{Y}}^2(\mathbf{Y}_i)} - \hat{I}_n^{IS}\right)^2. \tag{1.19}$$

Remark 1.2 *The variance of $\hat{I}_n^{IS}$ can be greater than the variance of the Monte Carlo estimator if $f_{\mathbf{Y}}$ is not well chosen The integral in Equation (1.18) can be rewritten as*

$$\begin{aligned}\int\left(H^2(\mathbf{y})\frac{f_{\mathbf{X}}(\mathbf{y})^2}{f_{\mathbf{Y}}(\mathbf{y})}\right)d\mathbf{y} &= \int\left(H^2(\mathbf{y})\frac{f_{\mathbf{X}}(\mathbf{y})}{f_{\mathbf{Y}}(\mathbf{y})}f_{\mathbf{X}}(\mathbf{y})\right)d\mathbf{y},\\ &= \mathbb{E}\left[H^2(\mathbf{X})\frac{f_{\mathbf{X}}(\mathbf{X})}{f_{\mathbf{Y}}(\mathbf{X})}\right].\end{aligned}$$

Therefore, the importance sampling estimator has a lower variance than the standard Monte Carlo estimator if

$$\mathbb{E}\left[H^2(\mathbf{X})\frac{f_{\mathbf{X}}(\mathbf{X})}{f_{\mathbf{Y}}(\mathbf{X})}\right] \leq \mathbb{E}\left[H^2(\mathbf{X})\right].$$

Moreover, if $H(\mathbf{Y})f_{\mathbf{X}}(\mathbf{Y})/f_{\mathbf{Y}}(\mathbf{Y}) \in \mathbb{L}^2$ it comes

$$\frac{\sqrt{n}}{\widehat{\sigma}_n^{IS}}\left(\hat{I}_n^{IS} - I\right) \underset{n\to+\infty}{\overset{\mathcal{L}aw}{\longrightarrow}} \mathcal{N}(0,1), \tag{1.20}$$

and an asymptotic confidence interval at $100(1-\alpha)\%$ is

$$CI_\alpha = \left[\hat{I}_n^{IS} - \Phi^{-1}(1-\alpha/2)\widehat{\sigma}^{IS}, \hat{I}_n^{IS} + \Phi^{-1}(1-\alpha/2)\widehat{\sigma}_n^{IS}\right]. \tag{1.21}$$

Equations above do not provide information about the best choice of an importance density. Minimising the variance given in (1.18), the method of Lagrange multipliers provides the optimal density $f_{\mathbf{Y}}$. Define

$$L(\lambda) = \int\left(H^2(\mathbf{y})\frac{f_{\mathbf{X}}(\mathbf{y})^2}{f_{\mathbf{Y}}(\mathbf{y})}\right)d\mathbf{y} - \lambda\left(\int f_{\mathbf{Y}}(\mathbf{y})d\mathbf{y} - 1\right).$$

Finding the zero of the derivative, it comes

$$H^2\frac{f_{\mathbf{X}}^2}{f_{\mathbf{Y}}} - \lambda f_{\mathbf{Y}} = 0,$$

then

$$f_{\mathbf{Y}} = \frac{|H|f_{\mathbf{X}}}{\sqrt{\lambda}},$$

with $\lambda > 0$. Under the constraint that $f_{\mathbf{Y}}$ is a probability density function, it must verify $\int f_{\mathbf{Y}}(\mathbf{y})du = 1$ and then

$$\sqrt{\lambda} = \int |H(\mathbf{y})|f_{\mathbf{X}}(\mathbf{y})d\mathbf{y}.$$

Finally, the probability density function which minimises the variance of (1.17) is equal to

$$f_{\mathbf{Y}}^* = \frac{|H|f_{\mathbf{X}}}{\int |H(\mathbf{y})|f_{\mathbf{X}}(\mathbf{y})d\mathbf{y}} = \frac{|H|f_{\mathbf{X}}}{\mathbb{E}[|H(\mathbf{X})|]}. \tag{1.22}$$

If H is a positive function, determining this optimal density is equivalent to know the quantity to be estimated.

1.4.2 Application to probability estimation

Assume now $H = \mathbb{1}_{\{g \leq 0\}}$. Equations (1.17) and (1.19) become respectively

$$\hat{I}_n^{IS} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{Y}_i) \leq 0\}} \frac{f_{\mathbf{X}}(\mathbf{Y}_i)}{f_{\mathbf{Y}}(\mathbf{Y}_i)}, \tag{1.23}$$

$$\left(\hat{\sigma}_n^{IS}\right)^2 = \frac{1}{n-1} \sum_{i=1}^{n} \left(\mathbb{1}_{\{g(\mathbf{Y}_i) \leq 0\}} \frac{f_{\mathbf{X}}^2(\mathbf{Y}_i)}{f_{\mathbf{Y}}^2(\mathbf{Y}_i)} - \hat{I}_n^{IS} \right)^2, \tag{1.24}$$

and the optimal probability density function is

$$f_{\mathbf{Y}} = \frac{\mathbb{1}_{\{g \leq 0\}} f_{\mathbf{X}}}{p}, \tag{1.25}$$

which depends on p.

Remark 1.3 *The estimator $\hat{I}_n^{IS}$ in (1.23) is unbiased if*

$$\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\} \subset supp(f_{\mathbf{Y}}).$$

An alternative to importance sampling is the *weighted importance sampling* estimator [10, 97]:

$$\hat{I}_n^{WIS} = \frac{\sum_{i=1}^{n} H(\mathbf{Y}_i) L(\mathbf{Y}_i)}{\sum_{i=1}^{n} L(\mathbf{Y}_i)}, \tag{1.26}$$

where $L = f_{\mathbf{X}}/f_{\mathbf{Y}}$. It can be useful to introduce these weights when the constant of normalisation of the importance density is difficult to obtain. Indeed, if

$$f_{\mathbf{Y}} = h_{\mathbf{Y}} / \int_{\mathbb{R}^d} h_{\mathbf{Y}}(\mathbf{y}) d\mathbf{y},$$

then (1.26) becomes

$$\hat{I}_n^{WIS} = \frac{\sum_{i=1}^{n} H(\mathbf{Y}_i) f_{\mathbf{X}}(\mathbf{Y}_i)/h_{\mathbf{Y}}(\mathbf{Y}_i)}{\sum_{i=1}^{n} f_{\mathbf{X}}(\mathbf{Y}_i)/h_{\mathbf{Y}}(\mathbf{Y}_i)}.$$

Finally, the estimator can be expressed as

$$\hat{I}_n^{WIS} = \sum_{i=1}^{n} W(\mathbf{Y}_i) H(\mathbf{Y}_i), \tag{1.27}$$

where $W(\mathbf{Y}_i) = L(\mathbf{Y}_i) / \sum_{j=1}^{n} L(\mathbf{Y}_j)$. This estimator is biased, but since the weights are in $[0, 1]$, the estimator $\hat{I}_n^{WIS}$ is comprised between the minimum and the maximum of H. Then, it is bounded which is not guaranteed by (1.17). In [45], conditions on the weights are given to have a consistent estimator with asymptotic normality. The choice of the importance density is widely studied in [19].

1.5 Conditional Monte Carlo

Conditional Monte Carlo is a variance reduction technique which consists in rewriting the standard Monte Carlo estimator by conditioning some variables. This method was generalized in [63]. Let $H : \mathbb{R}^d \to \mathbb{R}$ and $\mathbf{X}$ be a random vector in $\mathbb{R}^d$ such that it can be split in a couple of random vectors $(\mathbf{Y}, \mathbf{Z})$. Define $I = \mathbb{E}[H(\mathbf{X})] = \mathbb{E}[H(\mathbf{Y}, \mathbf{Z})]$ and conditioning according to, for example, $\mathbf{Z}$ it comes $I = \mathbb{E}[\mathbb{E}[H(\mathbf{Y}, \mathbf{Z})|\mathbf{Z}]]$. Let $(\mathbf{X}_n)_{n\geq 1} = ((\mathbf{Y}_n, \mathbf{Z}_n))_{n\geq 1}$ be an independent and identically distributed sequence of random vectors distributed as $\mathbf{X}$. A Monte Carlo estimator of I is

$$\hat{I}_n^{cond} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[H(\mathbf{Y}, \mathbf{Z}_i)|\mathbf{Z}_i]. \tag{1.28}$$

The variance of (1.28) can be compared with the standard Monte Carlo estimator

$$\begin{aligned}
\mathrm{Var}\left(\hat{I}_n^{cond}\right) &= \frac{1}{n}\mathrm{Var}(\mathbb{E}[H(\mathbf{Y}, \mathbf{Z})|\mathbf{Z}]), \\
&= \frac{1}{n}\left(\mathbb{E}(\mathbb{E}^2[H(\mathbf{Y}, \mathbf{Z})|\mathbf{Z}]) - \mathbb{E}^2[H(\mathbf{X})]\right), \\
&\leq \frac{1}{n}\left(\mathbb{E}(\mathbb{E}[H^2(\mathbf{Y}, \mathbf{Z})|\mathbf{Z}]) - \mathbb{E}^2[H(\mathbf{X})]\right), \\
&= \frac{1}{n}\left(\mathbb{E}[H^2(\mathbf{Y}, \mathbf{Z})) - \mathbb{E}^2[H(\mathbf{X})]\right), \\
&= \mathrm{Var}\left(\hat{I}_n^{MC}\right).
\end{aligned} \tag{1.29}$$

Equation (1.29) is obtained by Jensen inequality, recalled below, applied to the function $t \mapsto t^2$.

Definition 1.2 *(Jensen inequality). Let $f : J \subset \mathbb{R} \to \mathbb{R}$ be a convex functions and X a random variable taking its values in J. Then*

$$f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)].$$

Finally, this estimator has a lower variance than the one obtained from the standard Monte Carlo. The conditioning is efficient since the conditional expectation $\mathbb{E}[H(\mathbf{Y}, \mathbf{Z}_i)|\mathbf{Z}_i]$ is easy to compute. Otherwise, this expectation must be also estimated. In particular case, this expectation can be estimated without using standard Monte Carlo (see Section 1.7). Figure 1.4 provides an illustration of such a conditioning in dimension $d = 2$.

1.6 First and Second Order Reliability Method

Methods FORM/SORM (*First/Second Order Reliability Method*) are completely different from Monte Carlo methods and are dedicated to probability estimation. This class of methods was developed in a structural reliability context [77]. A general problem is to consider a structure that is subjected to a solicitation and its resistance. If these two quantities are represented by two random variables S and R, the probability of the event $\{R \leq S\}$ is a quantity of interest.

In this section, the estimation of such a probability does not use Monte Carlo (or importance sampling) techniques. The problem is seen as an optimisation problem and it is decomposed in three stages. The first one transforms the input random variables in independent standard

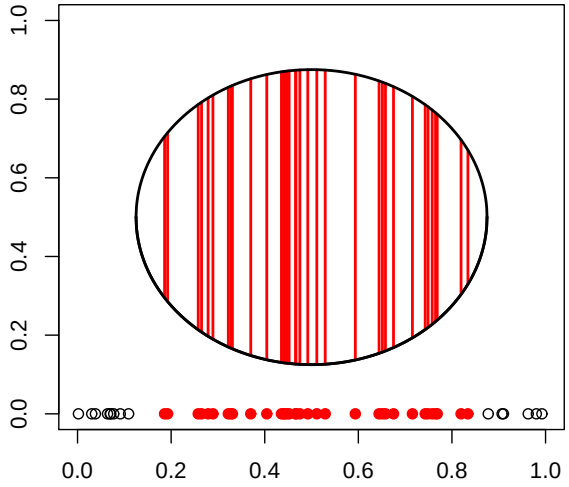


Figure 1.4: Illustration of conditional Monte Carlo in dimension $d = 2$. Assume that H takes its values in the disk and takes as input the random vector $\mathbf{X} = (X_1, X_2) \in [0,1]^2$. Red and black points represent a sample distributed as X_1. Each red line represents $\mathbb{E}[H(X_1, X_2)|X_1^i]$.

Gaussian random variables. The second one consists in getting the so-called *design point* on the limit state $\Gamma = \{\mathbf{x} \in \mathbb{U}, g(\mathbf{x}) = 0\}$, and approximating this surface by a hyperplane (resp. quadratic surface) for FORM (resp. SORM). The last stage consists in estimating p as the probability that the standard normal distribution is below the hyperplane (or the quadratic surface). These three stages are detailed in the following paragraphs.

Nonetheless, the estimation cannot be controlled and depends strongly on Γ and g. In practice, the design point can be used to create an importance density which provides a controllable estimator [84]. This construction is also described in Section 1.6.4.

1.6.1 Transformation to the Gaussian space

To simplify the development of some reliability methods, the random variables used as inputs of the numerical code are transformed to independent and identically distributed random variables. Such a transformation is used also in Section 1.7.

Rosenblatt's transformation [77] modified each component of $\mathbf{X}$ to a standard Gaussian random variable. This transformation is defined recursively by

$$\begin{aligned} T : \mathbb{R}^d &\longmapsto \mathbb{R}^d \\ \mathbf{x} = (x_1, \ldots, x_d) &\longrightarrow (T_1(x_1), \ldots, T_d(x_d)), \end{aligned} \tag{1.30}$$

with

$$\begin{aligned} T_1(x_1) &= \Phi^{-1}(F_{X_1}(x_1)) \\ T_i(x_i) &= \Phi^{-1}(F_{X_i}(x_i|x_1, \ldots, x_{i-1})) \quad \text{for } i = 2, \ldots, d, \end{aligned}$$

where F_{X_i} is the cumulative distribution function of X_i conditionally to X_j for $j < i$. When the random variables are independent, the transformation (1.30) becomes for $i = 1, \dots, d$

$$T_i(x_i) = \Phi^{-1}(F_{X_i}(x_i)),$$

where Φ^{-1} is the inverse of the function Φ. Let $\mathbf{Y} = T(\mathbf{X}) \sim \mathcal{N}_d(\mathbf{0}_d, I_d)$, the probability p can be rewritten as

$$p = \mathbb{P}\left(\left(g \circ T^{-1}\right)(\mathbf{Y}) \leq 0\right).$$

Alleviating the notation, until the end of this section, the function g is now equal to $g \circ T^{-1}$ and $\mathbf{X} \sim \mathcal{N}_d(\mathbf{0}_d, I_d)$ then $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$.

1.6.2 Searching for the design point

The key issue of this method is to obtain the most probable point in the limit state $\Gamma = \{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = 0\}$, called the design point. The design point is interpreted as the nearest point of the origin which belongs to the limit state Γ. It is the solution of the following optimisation problem

$$\begin{cases} \min\limits_{\mathbf{u} \in \mathbb{R}^d} \left(\mathbf{u}^t \mathbf{u}\right), \\ g(\mathbf{u}) = 0. \end{cases} \tag{1.31}$$

There exists several algorithms to solve Problem (1.31). These algorithms consists in getting, from a starting point $\mathbf{u}_0$, the best descent direction as well as the best distance to get a new point. From step (k) to $(k+1)$, define

$$\mathbf{u}_{(k+1)} = \mathbf{u}_{(k)} + \alpha_{(k)} S_{(k)}.$$

The following algorithms are based on gradient descent. They can be found in [77].

Hasofer-Lind-Rackwitz-Fiessler algorithm HLRF

It is an algorithm constructed by Hasofer-Lind-Rackwitz-Fiessler [64, 99] specific to Problem (1.31). Denote $\mathbf{u}^{(k)}$ obtained at step k. The point $\mathbf{u}_{(k+1)}$ is given by

$$\mathbf{u}_{(k+1)} = \mathbf{u}_{(k)} - \alpha_{(k)} \frac{g(\mathbf{u}_{(k)})}{\|\nabla g(\mathbf{u}_{(k)})\|}, \tag{1.32}$$

where $\alpha_{(k)} = \nabla g(\mathbf{u}_{(k)}) / \|\nabla g(\mathbf{u}_{(k)})\|$. Indeed, let $\mathbf{u}_{(k)}$ such that $g(\mathbf{u}_{(k)}) \leq 0$, the second order Taylor polynomial is

$$g(\mathbf{u}) = g(\mathbf{u}_{(k)}) + \nabla g(\mathbf{u}_{(k)})^t (\mathbf{u} - \mathbf{u}_{(k)}) + O(\mathbf{u} - \mathbf{u}_{(k)})^2.$$

The point $\mathbf{u}_{(k+1)}$ must verify:

$$g(\mathbf{u}_{(k+1)}) = g(\mathbf{u}_{(k)}) + \nabla g(\mathbf{u}_{(k)})^t (\mathbf{u}_{(k+1)} - \mathbf{u}_{(k)}) = 0,$$

dividing by the norm of the gradient, it comes

$$\frac{g(\mathbf{u}_{(k)})}{\|\nabla g(\mathbf{u}_{(k)})\|} + \frac{\nabla g(\mathbf{u}_{(k)})^t (\mathbf{u}_{(k+1)} - \mathbf{u}_{(k)})}{\|\nabla g(\mathbf{u}_{(k)})\|} = 0. \tag{1.33}$$

Set $\alpha_{(k)} = \nabla g(\mathbf{u}_{(k)})/\|\nabla g(\mathbf{u}_{(k)})\|$, Equation (1.33) becomes :

$$\frac{g(\mathbf{u}_{(k)})}{\|\nabla g(\mathbf{u}_{(k)})\|} + \alpha_{(k)}^t(\mathbf{u}_{(k+1)} - \mathbf{u}_{(k)}) = 0,$$

or

$$\alpha_{(k)}^t \mathbf{u}_{(k+1)} = \alpha_{(k)}^t \mathbf{u}_{(k)} - \frac{g(\mathbf{u}_{(k)})}{\|\nabla g(\mathbf{u}_{(k)})\|}.$$

Finally, since $\|\alpha_{(k)}\| = 1$ Equation (1.32) is verified.

The convergence of this algorithm is not assured since it highly depends on the limit state.

Remark 1.4 *There exists a refined version of the HLRF algorithm, which consists in finding at each step the optimal step size [113].*

Second order algorithm

For the second order algorithms, the Hessian of g must be computed. However, for high dimension it cannot be used in practice if g is time-consuming. Abdo-Rackwitz algorithm [77] is based on the rewriting of the Hessian matrix.

Remark 1.5 *All these algorithms require to start with an initial point $\mathbf{u}_0$. A possible choice is to take $\mathbf{u}_0 = \mathbf{0} \in \mathbb{R}^d$, the mean or a quantile of the inputs on the initial space.*

1.6.3 FORM/SORM approximation

In this subsection FORM and SORM approximations are considered. The first one estimates the limit state by a linear surface whereas the second uses a quadratic surface.

First order approximation

Assume $\mathbf{u}^*$ is a solution of Problem (1.31). FORM approximation method consists in approximating Γ by a hyperplane passing through $\mathbf{u}^*$. This hyperplane h is defined by

$$h(\mathbf{x}) = \nabla g(\mathbf{u}^*)^t(\mathbf{x} - \mathbf{u}^*).$$

The numerical code g is approximated around $\mathbf{u}^*$ by h, then the probability $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ is estimated by

$$\begin{aligned}\mathbb{P}(h(\mathbf{X}) \leq 0) &= \mathbb{P}(\nabla g(\mathbf{u}^*)^t(\mathbf{X} - \mathbf{u}^*) \leq 0) \\ &= \mathbb{P}\left(\frac{\nabla g(\mathbf{u}^*)^t \mathbf{X}}{\|\nabla g(\mathbf{u}^*)\|} \leq \frac{\nabla g(\mathbf{u}^*)^t \mathbf{u}^*}{\|\nabla g(\mathbf{u}^*)\|}\right).\end{aligned}$$

Notice that $\nabla g(\mathbf{u}^*)^t \mathbf{X}/\|\nabla g(\mathbf{u}^*)\| \sim \mathcal{N}(0,1)$, and denoting Φ the cumulative density function of a standard normal distribution, it comes

$$\mathbb{P}(h(\mathbf{X}) \leq 0) = \Phi(-\beta^*),$$

with $\beta^* = -\nabla g(\mathbf{u}^*)^t \mathbf{u}^*/\|\nabla g(\mathbf{u}^*)\|$.

Second order approximation

The SORM approximation depends on the curvature of the limit state $\Gamma = \{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = 0\}$. For a given solution $\mathbf{u}^*$ of Problem (1.31), denote $\beta = \|\mathbf{u}^*\|$. This quadratic approximation comes from [18] and needs the main curvatures $(\kappa_1, \ldots, \kappa_{d-1})$ of Γ at $\mathbf{u}^*$. The probability is then estimated by

$$\Phi(-\beta) \left(\prod_{j=1}^{d-1} (1 + \beta\kappa_j)^{-1/2} \right).$$

This method is called *asymptotic* since this approximation holds for $\beta \to +\infty$.

In practice, the search for the design point requires few calls to the function g, but a good precision of the approximation depends on g. Indeed, g must to be a differentiable function, have a linear or quadratic limit state, and have a unique design point. In [42], a method was developed if many design points exist. In [50], a criterion is established to check the uniqueness of the design point. However, the estimator cannot be controlled even if these hypotheses are verified.

1.6.4 FORM/SORM-Importance sampling

To overcome the lack of control of the estimator, it seems natural to couple the search for the design point with importance sampling techniques [84]. Indeed, assume that the solution of Problem (1.31) is $\mathbf{u}^*$. Then, simulating close to $\mathbf{u}^*$, there is a high probability to get simulations in $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$. Choosing the importance density as

$$f_{\mathbf{Y}}(\mathbf{u}) = \frac{1}{(2\pi)^{d/2}} \exp -\|\mathbf{u} - \mathbf{u}^*\|^2/2,$$

Equations (1.17) and (1.19) become respectively

$$\hat{p}_n^{F/S-IS} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{g(\mathbf{Y}_i) \leq 0\}} \exp\left(\|\mathbf{Y}_i\|^2/2 - \mathbf{Y}_i^t \mathbf{u}^* \right),$$

$$\hat{\sigma}_n^{F/S-IS} = \frac{1}{n-1} \sum_{i=1}^{n} \left(\mathbb{1}_{\{g(\mathbf{Y}_i) \leq 0\}} \exp\left(\|\mathbf{Y}_i\|^2/2 - \mathbf{Y}_i^t \mathbf{u}^* \right) - \hat{p}_n^{F/S-IS} \right)^2.$$

where $\mathbf{Y}_i \sim \mathcal{N}_d(\mathbf{u}^*, I_d)$. As for Monte Carlo or importance samplings methods, it can be deduced a confidence interval at level $100(1-\alpha)\%$

$$CI_\alpha = [\hat{p}_n^{F/S-IS} - \Phi^{-1}(1-\alpha/2)\hat{\sigma}_n^{F/S-IS}, \hat{p}_n^{F/S-IS} + \Phi^{-1}(1-\alpha/2)\hat{\sigma}_n^{F/S-IS}]. \tag{1.34}$$

1.7 Directional Sampling

1.7.1 General description

Conditional Monte Carlo methods require some information about the function H. If the conditional expectation is difficult to estimate (e.g. a standard Monte Carlo approach) the gain of this method will decrease. In the particular case where $H = \mathbb{1}_{\{g \leq 0\}}$ with g a black-box function, such methods cannot be used without modifications. The directional sampling method [13, 43] provides an estimation of such expectations. Assume that $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ with $\mathbf{X} = (X^1, \ldots, X^d)$

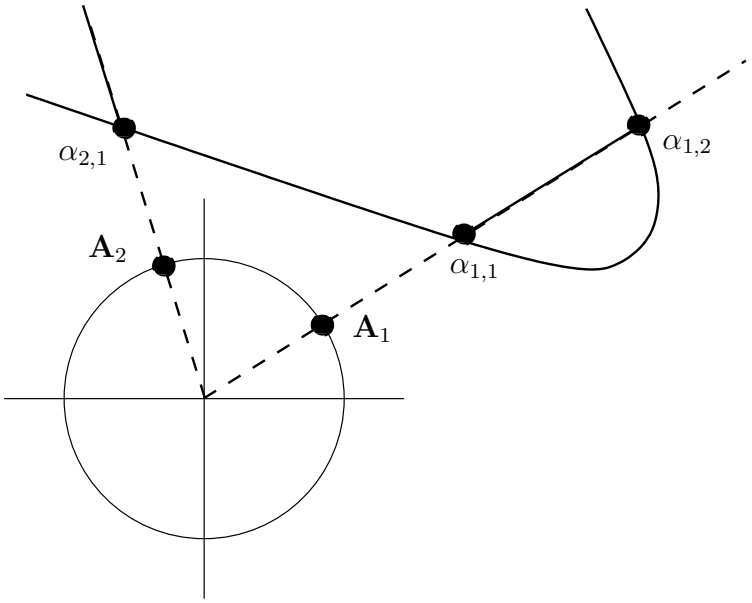


Figure 1.5: Directional sampling: representation of the method in dimension 2. The random vectors $\mathbf{A}_1, \mathbf{A}_2$ are uniformly distributed on $\mathbb{S}^2$. For $i = 1, 2$, once the solution of $g_i(R) = 0$ is found, the plain curve delimits the set $\{g_i(R) \leq 0\}$.

a random vector. Conditioning by $X^2, \ldots, X^d$ is equivalent to estimate $\mathbb{P}(g(\mathbf{X}) \leq 0 | X^2, \ldots, X^d)$ or to find the solutions of $g(X^1, X^2, \ldots, X^d) = 0$ knowing $X^2, \ldots, X^d$.

Simplifying the description, consider that the transformation T described in Section 1.6.1 has been applied to $\mathbf{X}$ and then $\mathbf{X} \sim \mathcal{N}_d(\mathbf{0}, I_d)$. Denote χ^2_d the chi-squared distribution with d degrees of freedom and $\mathbb{S}^d = \{\mathbf{x} \in \mathbb{R}^d, \|\mathbf{x}\| = 1\}$ the unit d-sphere. The random vector $\mathbf{X}$ can be defined as $R\mathbf{A}$ where $R \sim \chi^2_d$ and $\mathbf{A} \sim \mathcal{U}(\mathbb{S}^d)$ and then $p = \mathbb{P}(g(R\mathbf{A}) \leq 0) = \mathbb{E}[\mathbb{P}(g(R\mathbf{a}) \leq 0 | \mathbf{A} = \mathbf{a})]$. Since R is a real random variable, if $\mathbf{A}$ is known it remains to find the solutions of $g(R\mathbf{A}) = 0$.

Let $(\mathbf{A}_i)_{i \geq 1}$ be an iid sequence of random vectors uniformly distributed on $\mathbb{S}^d$. The conditional Monte Carlo method allows to estimate p as follow

$$\widehat{p}_n^{DS} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{P}(g(R\mathbf{A}_i) \leq 0 | \mathbf{A}_i).$$

This estimator has the same form than the one given in (1.28). If $\mathbb{P}(g(R\mathbf{A}_i) \leq 0 | \mathbf{A}_i)$ is known then its variance is lower than the variance of the Monte Carlo estimator. Otherwise, the quantity $\mathbb{P}(g(R\mathbf{A}_i) \leq 0 | \mathbf{A}_i)$ must be estimated.

Knowing $\mathbf{A}_i$, $g_i(R) = g(R\mathbf{A}_i)$ is a real valued function and estimate $\mathbb{P}(g(R\mathbf{A}_i) \leq 0 | \mathbf{A}_i)$ is equivalent to find the solutions of $g_i(R) = 0$. Denote $0 \leq \alpha_{i,1} \leq \cdots \leq \alpha_{i,m_i}$ the solutions of $g_i(R) = 0$, then

$$\mathbb{P}(g_i(R) \leq 0 | \mathbf{A}_i) = \sum_{j=1}^{m_i} (-1)^{j+1} (1 - F_{\chi^2_d}(\alpha_{i,j})),$$

where $F_{\chi^2_d}$ is the cumulative distribution function of a χ^2_d (see Figure 1.5). It remains to find for all i the solutions $g_i(R) = 0$ (e.g. using dichotomic algorithm or Newton-Raphson methods). Many methods of roots finding can be found in [89].

1.7.2 Adaptive Directional Sampling

This method was developed in [90] and combines stratified simulation with directional sampling. Denote $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ with $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, I_d)$. As said in Section 1.7.1, $\mathbf{X}$ can be rewritten as $R\mathbf{A}$ where $R \sim \chi^2_d$ and $\mathbf{A} \sim \mathcal{U}(\mathbb{S}^d)$. The first stage of this method consists in making a partition $S_1, \ldots, S_m$ of $\mathbb{S}^d$. Assume there are n simulations available to estimate p and $n_i = \lfloor n\omega_i \rfloor$ simulations drawn on S_i with $\sum_{i=1}^m \omega_i = 1$. The probability p can be defined as

$$p = \sum_{i=1}^{m} \mathbb{P}(\mathbf{A} \in S_i)\mathbb{E}[\mathbb{P}(g(R\mathbf{A}^i) \leq 0 | \mathbf{A}^i)],$$

where $\mathbf{A}^i$ is a random vector uniformly distributed on S_i. Let $(\mathbf{A}^i_j)_{j\geq 1}$ be an iid sequence of random vectors distributed as $\mathbf{A}^i$, p is estimated by

$$\widetilde{p}_n = \sum_{i=1}^{m} \mathbb{P}(\mathbf{A} \in S_i) \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbb{P}(g(R\mathbf{A}^i_j) \leq 0 | \mathbf{A}^i_j)],$$

where $\mathbb{P}(g(R\mathbf{A}^i_j) \leq 0 | \mathbf{A}^i_j)]$ is estimated as described in Section 1.7.1. The variance of $\widetilde{p}_n$ is equal to

$$\mathrm{Var}(\widetilde{p}_n) = \frac{1}{n} \sum_{i=1}^{m} \frac{\rho_i^2 \sigma_i^2}{\omega_i}, \tag{1.35}$$

with

$$\begin{aligned} \rho_i &= \mathbb{P}(\mathbf{A} \in S_i), \\ \sigma_i &= \mathrm{Var}(\mathbb{P}(g(R\mathbf{A}^i) \leq 0 | \mathbf{A}^i)). \end{aligned}$$

In this construction, the value of ω_i must be estimated. Minimising the variance of $\widetilde{p}_n$, the optimal allocations are given by

$$\omega_i = \frac{\rho_i \sigma_i}{\sum_{j=1}^{m} \rho_j \sigma_j},$$

and (1.35) becomes

$$\mathrm{Var}(\widetilde{p}_n) = \left(\sum_{i=1}^{m} \rho_i \sigma_i \right)^2. \tag{1.36}$$

Two estimators based on this method can be defined: the *non-recycling* I^{nr} and *recycling* I^r adaptive directional stratified estimator. These estimators are expressed as

$$\begin{aligned} I^r &= \sum_{i=1^m} \rho_i \frac{1}{N_i^r} \sum_{j=1}^{N_i^r} \mathbb{P}(g(R\mathbf{A}^i_j) \leq 0 | \mathbf{A}^i_j), \\ I^{nr} &= \sum_{i=1^m} \rho_i \frac{1}{N_i^{nr}} \sum_{j=1}^{N_i^{nr}} \mathbb{P}(g(R\mathbf{A}^i_{n_i+j}) \leq 0 | \mathbf{A}^i_{n_i+j}). \end{aligned}$$

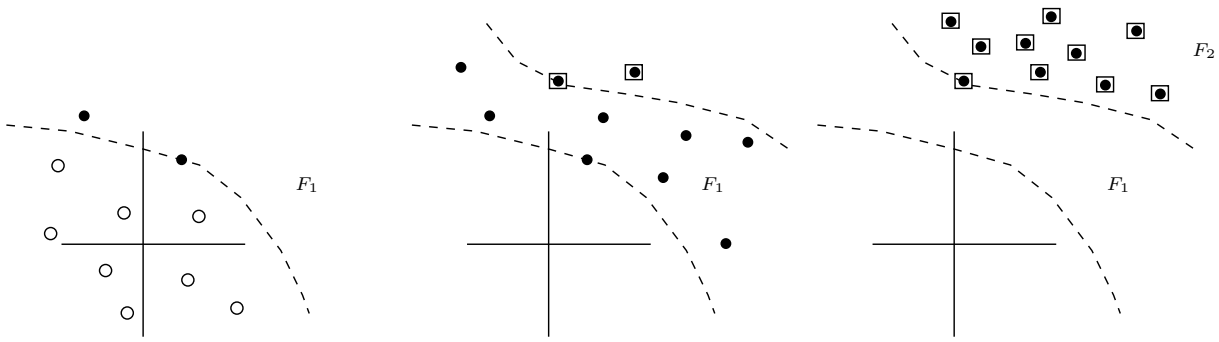


Figure 1.6: Illustration of the splitting method. **Left**: A first sample is simulated according to the initial distribution of $\mathbf{X}$. A first subset F_1 is determined from the sample (black points). **Middle**: The black points are distributed as $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_1)$. As for the first step, this sample defines a new subset F_2. **Right**: The last step consists in simulating according to $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_2)$.

The recycling estimator uses the sample simulated on each strata S_i to estimate the probability p. The non recycling estimator uses new simulations independent of the simulations drawn in the first step.

These two estimators have several advantages and shortcomings. The non-recycling estimator requires less simulations than the recycling one, but suffers from a bias which can be difficult to estimate. The recycling estimator requires more runs of the numerical code g, but is unbiased which can be more important for some studies. Finally, for these two estimators, a central limit theorem was provided also in [90].

1.8 Multilevel method

Multilevel splitting methods is a general class of methods which takes benefit of describing $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ as a product of greater probabilities. It was developed in [25, 61] and in [3] under the name *Subset Simulation*. This class of methods was also studied in [74] where the model is represented by a branching process. These methods are detailed in the following paragraph and a general illustration is provided on Figure 1.6.

1.8.1 Subset Simulation

Let F be a set containing $\{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq 0\}$, for example $F = \{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq S\}$ with $S > 0$. Then $p = \mathbb{P}(g(\mathbf{X}) \leq 0) = \mathbb{P}(g(\mathbf{X}) \leq 0)|\mathbf{X} \in F)\mathbb{P}(\mathbf{X} \in F)$. Since these two quantities are respectively greater than p, it will be easier to estimate each of them (e.g. by a standard Monte Carlo). If the conditional probability is still too small to be easily estimated, this step can be repeated with a sequence of sets $F_1, \dots F_n$ such that

$$\mathbb{R}^d = F_0 \supset F_1 \supset \dots \supset F_{n-1} \supset F_n = \{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq 0\},$$

and by induction:

$$p = \prod_{k=1}^{n} \mathbb{P}(\mathbf{X} \in F_k|\mathbf{X} \in F_{k-1}). \tag{1.37}$$

For instance assume that the sets $(F_k)_{k\geq 1}$, and the value of n are known as well as simulating according to $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_{k-1})$. For all $k \geq 1$, the probability $p_k = \mathbb{P}(\mathbf{X} \in F_k|\mathbf{X} \in F_{k-1})$ can be

estimated by standard Monte Carlo

$$\widetilde{p}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \mathbb{1}_{\left\{\mathbf{X}_i^{(k-1)} \in F_k\right\}}, \tag{1.38}$$

where $\left(\mathbf{X}_i^{(k-1)}\right)_{i\geq 1}\Big)$ is a sequence of iid random vectors distributed as $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_{k-1})$. Once each p_k is estimated according to (1.38), p is estimated by

$$\widehat{p}_n^{Sub} = \prod_{k=1}^{n} \widetilde{p}_k.$$

The choice of each F_k can be made using a decreasing sequence of real values $(S_k)_{k\geq 1}$ such that $S_n = 0$ and for all $1 \leq k \leq n, F_k = \{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq S_k\}$. The value of S_k must be well chosen since this choice influences the number of steps n. Indeed, if estimating each p_k needs a few simulations, it implies that p_k is largely greater than p and the number n of steps is large. If n is too small this implies that p_k is small too and it will be difficult to estimate it. It is proposed in [3] to choose adaptively each S_{k+1} as one of the quantile of the random variable $g\left(\mathbf{X}^{(k-1)}\right)$.

Assume now that $p = 10^{-n}$. Using a standard Monte Carlo sampling it is necessary to make approximately 10^{n+2} calls to the numerical code to get a coefficient of variation equal to 10%. Assume besides that the subset simulation method can split the probability p in n probability equal to 10^{-1}. Estimating all of these probabilities, with a coefficient of variation equal to 10%, needs approximately $n \times 10^3$ calls to the numerical code, which is lower than 10^{n+2} for all n.

How to simulate according to the conditional law?

In (1.38) it is assumed that a sample distributed as $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_{k-1})$ is easy to obtain. In practice, when the numerical code is a black box, such information is generally unknown. The original Metropolis algorithm consists in simulating according to a so-called instrumental distribution. Then, the produced sample is tested if it could be sample according to the target distribution. This approach can be ineffective when the dimension increases. A modified Metropolis algorithm is provided in [3] which reduces the influence of the dimension. These two algorithms can be summarised in two steps : 1) generate a candidate according to the condition to be, at step k, in F_{k-1}; 2) accept this candidate if it is in F_k.

Instead of simulating a candidate $\mathbf{x} = (x_1, \ldots, x_d)$, the modified Metropolis algorithm simulates independently each component x_j of a candidate. This type of simulation is similar to the Gibbs sampling. This algorithm is described below. Let $\mathbf{x} = (x_1, \ldots, x_d)$ and $f(.|\mathbf{x}) = \prod_{j=1}^d f_j(.|x_j)$ be the proposal probability density function centred in $\mathbf{x}$ and assume it is symmetrical.

1.8.2 Splitting method

The splitting method, developed in [25, 61], provides an estimation of a rare event probability. As said in Section 1.8.1, choosing the values S_k can be difficult. The idea of this method consists in choosing the "worst" simulation obtained at each step. In other words, at step 1 let $\mathbf{X}_1^{(1)}, \ldots, \mathbf{X}_N^{(1)}$ be a sample distributed as $\mathbf{X}$. The first threshold is given by $S_1 = \max(g(\mathbf{X}_1^{(1)}), \ldots, g(\mathbf{X}_N^{(1)}))$. The sample which defines the threshold S_1 is simulated according to $\mathcal{L}(\mathbf{X}|g(\mathbf{X}) \leq S_1)$. At step k, let $\mathbf{X}_1^{(k)}, \ldots, \mathbf{X}_N^{(k)}$ be a sample distributed as $\mathcal{L}(\mathbf{X}|g(\mathbf{X}) \leq S_{k-1})$. As for the first step, a

Algorithm 1.1 Modified Metropolis algorithm

$k \geq 1$; $\mathbf{X}_1^{(k-1)}, \ldots, \mathbf{X}_i^{(k-1)}$
1) Generate a candidate $\widetilde{\mathbf{X}} = (\widetilde{X}_1, \ldots, \widetilde{X}_d)$:
for $j = 1, \ldots, d$ **do**
simulate $\widetilde{X}_j$ according to $f_j(.|X_{i,j}^{(k-1)})$
Let $r_j = f_j(\widetilde{X}_j)/f_j(X_{i,j}^{(k-1)})$ and

$$\tilde{X}_j = \begin{cases} \tilde{X}_j & \text{with probability } min(1, r_j) \\ X_{i,j}^{(k-1)} & \text{with probability } 1 - min(1, r_j) \end{cases}$$

end for
1) Accept or reject $\widetilde{\mathbf{X}}$ as $\mathbf{X}_{i+1}^{(k-1)}$:
if $\widetilde{\mathbf{X}} \in F_k$ **then**
$\mathbf{X}_{i+1}^{(k-1)} = \widetilde{\mathbf{X}}$
else
$\mathbf{X}_{i+1}^{(k-1)} = \mathbf{X}_i^{(k-1)}$
end if

new threshold is defined by $S_k = \max(g(\mathbf{X}_1^{(k)}), \ldots, g(\mathbf{X}_N^{(k)}))$. The "worst" simulation is then simulated according to $\mathcal{L}(\mathbf{X}|g(\mathbf{X}) \leq S_k)$. Let $k \geq 1$, knowing the threshold S_k, there is a proportion $1 - 1/N$ of the sample $\mathbf{X}_1^{(k)}, \ldots, \mathbf{X}_N^{(k)}$ which verified $g(.) \leq S_k$. The probability $\mathbb{P}(g(\mathbf{X}) \leq S_k | g(\mathbf{X}) \leq S_{k-1})$ is estimated by $1 - 1/N$. The algorithm is defined below.

Algorithm 1.2 Splitting algorithm

Set $N \geq 1$
Let $\mathbf{X}_1^{(1)}, \ldots, \mathbf{X}_N^{(1)}$ be a sequence of iid random vectors distributed as $\mathbf{X}$
for $k = 1, 2, \ldots$ **do**
set $S_k = \max\left(g(\mathbf{X}_1^{(k)}), \ldots, g(\mathbf{X}_N^{(k)})\right)$
for $i = 1, \ldots, N$ **do**

$$\mathbf{X}_i^{(k+1)} = \begin{cases} \mathbf{X}_i^{(k)} & \text{if } g(\mathbf{X}_i^{(k)}) < S_k \\ \mathbf{X}^{(k+1)} \sim \mathcal{L}(\mathbf{X}|g(\mathbf{X}) < S_k) & \text{if } g(\mathbf{X}_i^{(k)}) = S_k \end{cases}$$

end for
end for
Stop algorithm at step K when $S_K \geq 0$ and $S_{K+1} < 0$

Repeating the algorithm on K steps, the probability p is estimated by

$$\hat{I}^{Splitt} = \left(1 - \frac{1}{N}\right)^K,$$

where K is distributed as a Poisson distribution of parameter $(-N \log p)$. In [61], it is assumed, in a first description of the algorithm, the conditional sample can be made perfectly with no more calls to the numerical code and is called the *idealized algorithm*. Since, the $\hat{I}^{Splitt}$ takes

discrete values : $\left((1-1/N)^k\right)_{k\geq 1}$, it comes

$$\mathbb{P}\left(\hat{I}^{Splitt} = (1-1/N)^k\right) = \mathbb{P}(K=k) = \frac{p^N(-N\log p)^k}{k!},$$

and its variance is equal to

$$\mathrm{Var}\left(\hat{I}^{Splitt}\right) = p^2(p^{-1/N}-1).$$

Let $\alpha \in [0,1]$, a $100(1-\alpha)\%$ confidence interval is $[CI_\alpha^-, CI_\alpha^+]$ where

$$CI_\alpha^\pm = \hat{I}^{Splitt}\exp\left\{\pm\frac{\Phi^{-1}(1-\alpha/2)}{\sqrt{N}}\sqrt{-\log\hat{I}^{Splitt} + \frac{\left(\Phi^{-1}(1-\alpha/2)\right)^2}{4N}} - \frac{\left(\Phi^{-1}(1-\alpha/2)\right)^2}{2N}\right\}.$$

1.8.3 Importance splitting method

This approach [74, 75] describes the splitting techniques as a branching process. Remind that

$$p = \mathbb{P}(g(\mathbf{X})\leq 0) = \prod_{k=1}^{M+1}\mathbb{P}(\mathbf{X}\in F_k|\mathbf{X}\in F_{k-1}) = \prod_{k=1}^{M+1} p_k$$

with $\mathbb{R}^d = F_0 \supset F_1 \supset \cdots \supset F_{M+1} = \{\mathbf{x}\in\mathbb{R}^d,\ g(\mathbf{x})\leq 0\}$. Let us start with N random vectors distributed as $\mathbf{X}$. The first subset can be defined as $F_1 = \{\mathbf{x}\in\mathbb{R}^d, g(\mathbf{x})\leq q_1\}$ where q_1 is a quantile of the sample. Then, there is a proportion L_1/N that reaches F_1 and p_1 is estimated by $\hat{p}_1 = L_1/N$. Each of this L_1 simulations is replicated R_1 times and distributed as $\mathcal{L}(\mathbf{X}|\mathbf{X}\in F_1)$. At step 2, there are L_1R_1 simulations, and a proportion $L2/(L_1R_1)$ of the total number of simulations reaches the set F_2. As for p_1, the probability p_2 is estimated by $\hat{p}_2 = L_2/(L_1R_1)$. Repeating $M+1$ times the algorithm, the probability p is estimated by

$$\hat{I}^{branching} = \hat{p}_1\cdots\hat{p}_{M+1} = \frac{L_{M+1}}{NR_1\cdots R_M},$$

with L_{M+1} the number of simulations in the set $F_{M+1} = \{\mathbf{x}\in\mathbb{R}^d,\ g(\mathbf{x})\leq 0\}$ (see Figure 1.7). The estimator $\hat{I}^{branching}$ has no bias and its variance is equal to

$$\mathrm{Var}(\hat{I}^{branching}) = \frac{p^2}{N}\sum_{k=0}^{M}\frac{1}{R_1\cdots R_k}\left(\frac{1}{p_1\cdots p_{k+1}} - \frac{1}{p_1\cdots p_k}\right).$$

Let h be a function which represents a cost of simulation depending on the transition probability, then the average cost of the algorithm is equal to

$$C = N\sum_{k=0}^{M}\left(R_1\cdots R_k p_1\cdots p_k h(p_{k+1})\right).$$

It is shown in [74] that the parameters which minimise the variance $\hat{I}^{branching}$ are given by

$$\begin{cases} M &= \lfloor\frac{\log p}{y_0}\rfloor \\ p_i &= p^{1/(M+1)} \\ R_i &= p^{-1/(M+1)} \\ N &= \frac{C}{(M+1)h(P^{1/(m+1)})} \end{cases},$$

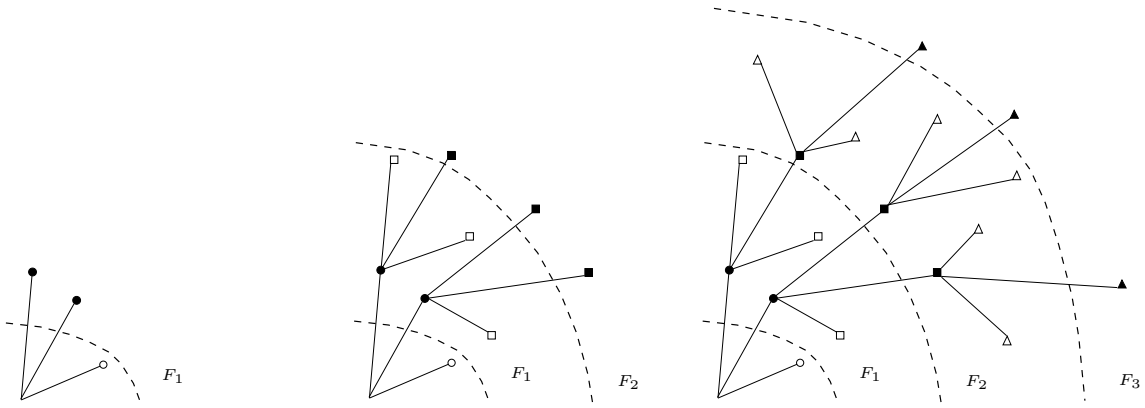


Figure 1.7: Illustration of the importance splitting method. **Left**: circles represent a sample distributed as $\mathbf{X}$. Black dots represent the simulations which have reach the set F_1. **Middle**: squares represent the replications of black circles and are distributed according to $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_1)$. Black squares represent the simulations which have reach the set F_2. **Right**: triangles represent a sample distributed as $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_2)$.

where y_0 is solution of the following equation

$$(2(1-e^y)+y)h(e^y) - y(1-e^y)e^y h'(e^y) = 0.$$

In the particular case $h = 1$ (like the Idealized Last Particle Algorithm seen in Section 1.8.2), it comes $M = \lfloor -0.6275 \log p \rfloor, R_i \approx 5$ and $p_i \approx 1/5$.

1.9 Sequential Monte Carlo

Standard Monte Carlo does not use the previous simulations to learn about the rare event. Sequential Monte Carlo can be seen as a generalisation since the current sample depends on the past. Recall that $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ can be estimated by $\frac{1}{n}\sum_{k=1}^n \mathbb{1}_{\{g(\mathbf{X}_k)\leq 0\}}$, where $(\mathbf{X}_k)_{k\geq 1}$ is an iid sequence of random vectors distributed as $\mathbf{X}$. Denoting $\mathcal{F}_k = \sigma\{\mathbf{X}_j,\ 1 \leq j \leq k\}$. Assume now that $\mathbf{X}_k$ is $(\mathcal{F}_{k-1})$-measurable with probability density function f_{k-1}. Let h be a function, f be the probability density function of $\mathbf{X}$ and denote $I = h(\mathbf{X})$. A general importance sampling estimator is provided by

$$I_n = \frac{1}{n}\sum_{k=1}^n h(\mathbf{X}_k)\frac{f(\mathbf{X}_k)}{f_{k-1}(\mathbf{X}_k)}. \tag{1.39}$$

This estimator has no bias if for all $k \geq 1$, $supp(f) \subset supp(f_{k-1})$. In the case where $h = \mathbb{1}_{\{g\leq 0\}}$, the conditions becomes $\{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq 0\} \subset supp(f_{k-1})$ for all $k \geq 1$. It comes

$$\begin{aligned}
n\mathbb{E}[I_n] &= \mathbb{E}\left[\sum_{k=1}^{n-1} h(\mathbf{X}_k)\frac{f(\mathbf{X}_k)}{f_{k-1}(\mathbf{X}_k)} + \mathbb{E}\left[h(\mathbf{X}_n)\frac{f(\mathbf{X}_n)}{f_{n-1}(\mathbf{X}_n)}\middle|\mathcal{F}_{n-1}\right]\right], \\
&= \mathbb{E}\left[\sum_{k=1}^{n-1} h(\mathbf{X}_k)\frac{f(\mathbf{X}_k)}{f_{k-1}(\mathbf{X}_k)}\right] + I, \\
&= (n-1)\mathbb{E}[I_{n-1}] + I,
\end{aligned}$$

and by induction $\mathbb{E}[I_n] = I$.

As in Section 1.4, the importance density can be difficult to build in practice. The importance density which minimises the variance of the estimator depends on the original probability density function and the quantity of interest. In [82], weighted sampling is introduced with the aim to be proportional to this ideal probability density function. In [27], this estimator is modified with the aim to obtain a central limit theorem which is generalised in [45].

1.10 Meta-modelling techniques

All methods provided in previous sections consider that only the numerical code g should be used to simulate a rare event. Statistical methods can be used to estimate the set $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$ or the function g. Once such an estimation is conducted, it becomes easier to simulate close to the set of interest $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$. Such estimators are called *meta-models* since the numerical code g is already a model of a real physical phenomenon. For example, FORM/SORM are based on a meta-model since they use an approximation of $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = 0\}$.

1.10.1 ^{2}SMART

The method ^{2}SMART [15] consists in rewriting the probability as in (1.37) where each threshold is determined by a meta-model which reproduces the value of g. Recall that $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ with $\mathbf{X} \sim \mathcal{N}_d(\mathbf{0}, I_d)$. As seen in Section 1.8.1, this probability can be expressed as

$$p = \prod_{k=1}^{n} \mathbb{P}(\mathbf{X} \in F_k | \mathbf{X} \in F_{k-1}),$$

with $\mathbb{R}^d = F_0 \supset F_1 \supset \cdots, \supset F_n = \{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq 0\}$. The main difficulties encounter in subset simulation is to find each subset F_k and to simulate according to $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_k)$. Let $(S_k)_{k\geq 1}$ be a decreasing sequence of real values, and consider now that each subset can be defined as $F_k = \{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq S_k\}$. Estimating the probability $p_k = \mathbb{P}(\mathbf{X} \in F_k | \mathbf{X} \in F_{k-1})$ can be time-consuming (requires a lot of runs to numerical code g) if S_k is not well chosen.

The method ^{2}SMART uses *Support Vector Machine* (SVM) [110], a learning tool which can be used as binary classifier or as regression model that mimics the numerical code g. The use of this meta-model helps to determine each threshold S_k, to simulate according to $\mathcal{L}(\mathbf{X}|\mathbf{X} \in F_{k-1})$ and to estimate each probability p_k. It is composed by three main stages.

For the first one, an initial sample is drawn and a first threshold S_1 (e.g. the quantile at 0.1% of the sample) is deduced. Then, a SVM is built to estimate the sets $\{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) = S_1\}$ and $F_1 = \{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) \leq S_1\}$. For the second one, a second SVM is built from a sample made on the margin of the estimation of $\{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) = S_1\}$ (determined by the first SVM). This strategy improves the accuracy of the estimation of $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = S_1\}$. For the last stage: using this second SVM, the estimation of p_1 can be obtained by the standard Monte Carlo. Finally, a new sample is made conditionally to be in the estimation of F_1.

Finally, these steps are repeated until the current threshold is lower or equal to 0. Let n be the number of steps. Each conditional probability p_k is estimated by $\widetilde{p}_k$ and p is then estimated by

$$\hat{p}^{(^2SMART)} = \prod_{k=1}^{n} \widetilde{p}_k.$$

For instance, neither the convergence of this estimator neither its asymptotic normality has not been proved.

1.10.2 Gaussian process based probability estimation

In this section, two probability estimation methods, based on Gaussian process meta-modelling (kriging) are summarised. The first one approximates the optimal probability density function of importance sampling. The second one provides a sequential design of experiments to estimate a rare event. Before continuing, the construction of Gaussian process meta-modelling, which becomes one of the most powerful meta-modelling techniques, is quickly recalled.

Consider that g is the realisation of a Gaussian stochastic process G. It takes the following form

$$G(\mathbf{x}) = H^t(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}),$$

where $H^t(\mathbf{x})\boldsymbol{\beta}$ is the deterministic part of the meta-model and Z is is its stochastic part defined by a Gaussian process with a zero-mean and and a covariance function $K(.,.)$. When Z is assumed to be stationary, its covariance function is rewritten as follow

$$K(\mathbf{x}, \mathbf{y}) = C(|\mathbf{x} - \mathbf{y}|, \boldsymbol{\theta}),$$

where $\boldsymbol{\theta}$ is a vector of parameters to be estimated. The estimation of these parameters is not detailed here, for more precisions see [105].

Consider a design of experiments $D_n = (\mathbf{x}_i, g(\mathbf{x}_i))_{1 \leq i \leq n}$ and let $\mathbf{x}_0$ not in $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Knowing D_n, the function G can be estimated by

$$\widehat{G}(\mathbf{x}_0) \sim \mathcal{N}(\widehat{m}(\mathbf{x}_0), \widehat{\sigma}^2(\mathbf{x}_0)),$$

where $\widehat{m}$ and $\widehat{\sigma}$ are two functions to be estimated which depends on $\boldsymbol{\theta}$. The main advantage of Gaussian processing is that the meta-model interpolates the design of experiments. In other words for all $i = 1, \ldots, n, G(\mathbf{x}_i) = g(\mathbf{x}_i)$

For this section, the following function is useful to estimate the probability p. Denote

$$\pi_n(\mathbf{x}) = \Phi\left(\frac{0 - \widehat{m}(\mathbf{x})}{\widehat{\sigma}(\mathbf{x})}\right),$$

where Φ represents the cumulative distribution function of standard Gaussian random variable.

Meta-modelling and importance sampling techniques

The importance density defined in (1.25) is intractable since it depends on the probability p. Moreover, simulating according to the importance distribution is not feasible in practice since the set $\{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$ is unknown. In [47], Gaussian processes are used to build a meta-model $\widehat{g}$ of g then to approximate the ideal importance density given in (1.25), estimated by

$$\widehat{f} = \frac{\mathbb{1}_{\{\widehat{G} \leq 0\}} f_{\mathbf{X}}}{p_\varepsilon},$$

where

$$p_\varepsilon = \int_{\mathbb{R}^d} \pi_n(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

Let $\mathbf{Y}$ be a random vector with probability density function $\widehat{f}$. The probability p is rewritten as follow

$$p = p_\varepsilon \mathbb{E}\left[\frac{\mathbb{1}_{\{g(\mathbf{Y}) \leq 0\}}}{\pi_n(\mathbf{Y})}\right].$$

The expectation term is a correction factor that represents the accuracy of the meta-model to predict the value of g: it is equal to 1 if $\widehat{g} = g$. Finally, the two quantities p_ε and the correction factor are both estimated by a Monte Carlo estimator. The estimation of the correction factor requires to simulate according to the estimation of the optimal density function. It is suggested in [47] to use Markov Chain Monte Carlo (MCMC) sampling to do it.

Nonetheless, this method does not take into account the previous simulations to update the meta-model by updating the estimation of hyper-parameters. In the following subsection, a sequential design of experiments is built to estimate rare event probabilities.

Stepwise Uncertainty Reduction (SUR) for probability estimation

Sequential methods seem to be more appropriate to estimate failure probabilities. As for [2]SMART methods, a meta-model is refined at each step of an algorithm to increase the accuracy of a probability estimator. In this section, the method proposed in [8], which uses Gaussian-process meta-modelling to build a sequential design of experiments, is briefly recalled. In [8] it is said that such criteria are especially efficient for probability estimation when Gaussian-process meta-modelling is used.

A design is produced to build a first Gaussian-process meta-model. At any following step, a set of points arising from the initial distribution is sampled to be a candidate for new evaluated point. Choosing a new point depends on a SUR criterion sampling defined by

$$J(\mathbf{x}) = \mathbb{E}\left[(p - \widehat{p}_{n+1})^2 \,|\mathbf{X}_{n+1} = \mathbf{x}\right],$$

where $\widehat{p}_{n+1} = \int_{\mathbb{R}^d} \pi_{n+1}(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$ and π_{n+1} built from the previous evaluated points. Then, the meta-model is updated and the procedure is repeated until the budget of run of the numerical code g is used. The SUR criterion cannot be used in practice but it is replaced by an adapted form which uses the design of experiments and the meta-model. Finally, the probability is estimated as follow

$$\widehat{p}_n = \int_{\mathbb{R}^d} \pi_n(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}.$$

1.11 Conclusion

An aspect almost not evoked in this chapter is the usability of these methods in function of the allowed number of runs of the numerical code and the dimension of the inputs. For example, methods based on meta-modelling become difficult to use in practice when the size of the design of experiment increases. The efficiency of FORM/SORM depends on the dimension. If the derivative of g is approximated, $2d$ runs of the numerical code are required at each step of the search for the design point. Moreover, they can be applied only if the derivative of the numerical code exists. The low-discrepancy sequences suffer from dimension also, pattern can appear in high dimension (see [79], Chapter 5). Moreover, Quasi-Monte Carlo estimator cannot be controlled.

Many constraints should definitely guide the choice of one or several techniques: the dimension, the number of runs allowed, the hypothesis on the numerical code and the control of the estimator. In practice, it is difficult to compare these methods since each of them can be used exclusively in a particular case of constraints. For example, the use of standard Monte Carlo is a good choice if the numerical code is not time-consuming, if the dimension is very high and

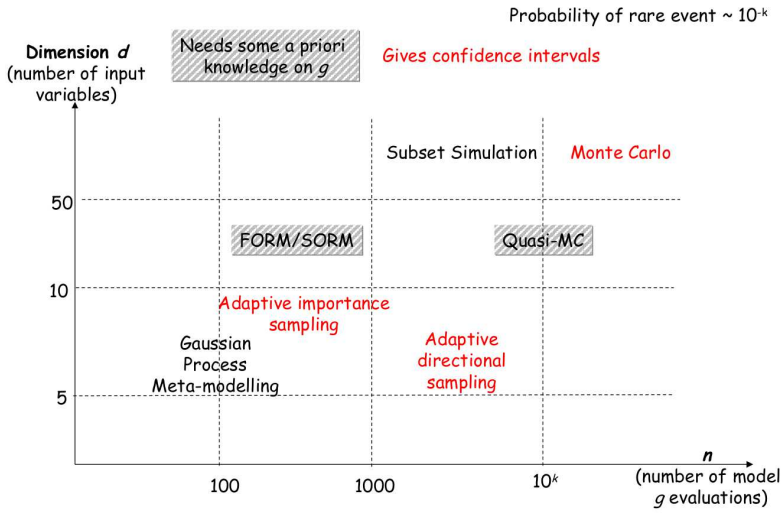


Figure 1.8: Methods presented in this chapter are placed in function of the dimension and the require number of runs to the numerical code. This figure was graciously provided by Bertrand Iooss.

if a confidence interval is required. In Figure 1.8, the methods are placed in function of the dimension and the number of runs of the numerical code needed for the estimation.

If regularity constraints (e.g. continuity) are often underlying evoked, they can be difficult to check in the reliability industrial computer codes. Discontinuity (edge effects) can appears (e.g. discontinuities of physics) that limits the varieties of possible approaches. To be liberate from these constraints, the numerical code can be considered binary, for example, it represents a decision rule.

The monotony is partially exploited in [89] but requires to calibrate many parameters (see Section 1.7.2). Moreover, monotonic constraints become to be a topic of interest. For example, for meta-modelling in [33] or for probability estimation in [100].

Chapter 2

Adaptation of classical methods to rare event estimation under monotonicity constraint

Résumé On introduit dans ce deuxième chapitre les hypothèses de monotonie ansi que les avantages qu'elles apportent. Après une présentation des méthodes existantes pour l'estimation de probabilité, on s'intéresse à l'adaptation de méthodes classiques d'estimation sous les hypothèses de monotonie.

Abstract The monotonic hypothesis is introduced in this second chapter and its immediate benefits are examined. After a description of existing methods for probability estimation, the adaptation of classical methods under the monotonicity hypothesis is carried out.

2.1 Introduction

In general for probability estimation studies, the numerical code g is considered black-box. Nonetheless, the physical phenomenon associated to g is in general totally or partially known. In such situations, some hypotheses on the behaviour of g can be made. In this thesis, the monotonic hypothesis is studied. Recall the intuitive idea that leads to such constraints. If a configuration leads to a safe (resp. undesirable) event, it is reasonable to say that a less (resp. more) restrictive configuration leads also to a safe (resp. undesirable) event.

The following section formalises this property in a more general case. Taking into account the monotonicity, the method developed in [16] to estimate a probability is detailed. In the last sections of this chapter, classical method (Standard Monte Carlo, splitting methods) are adapted with the aim to make the best use of monotonicity.

2.2 Introduction of monotonicity constraints

In this first section, the benefits of monotonic properties for probability estimation are provided. First, recall the definition of a globally monotonic function.

Definition 2.1 *Let $g : \mathbb{U} \subset \mathbb{R}^d \to \mathbb{R}$, g is said globally monotonic if g is monotonic relatively to x_i for all $i = 1, \dots, d$.*

Definition 2.2 *Let* $\mathbf{x} = (x_1, \ldots, x_d), \mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$ *such that for all* $i = 1, \ldots, d,\ x_i \leq y_i$. *This relation is the partial order and is denoted by* $\mathbf{x} \preceq \mathbf{y}$.

As seen in Sections 1.6 and 1.7, it can be more convenient to transform the input space. To facilitate the understanding the function g is transformed in a globally increasing function by the following mechanism. Denote $\mathbf{X} = (X_1, \ldots, X_d)$ the input random vector with independent components. To have a globally increasing function it is required to transform $\mathbf{X}$ almost as in Section 1.6.1. Let F_i be the cumulative distribution function of X_i and denote

$$\begin{aligned} T : \mathbb{R}^d &\longmapsto \mathbb{R}^d \\ \mathbf{x} = (x_1, \ldots, x_d) &\longrightarrow (T_1(x_1), \ldots, T_d(x_d)), \end{aligned} \tag{2.1}$$

with

$$T_i(x_i) = \begin{cases} x_i \leftarrow F_i(x_i) & \text{if } g \text{ increasing relatively to } x_i \\ x_i \leftarrow 1 - F_i(x_i) & \text{if } g \text{ decreasing relatively to } x_i \end{cases}.$$

Hence, the function $g \circ T : [0,1]^d \to \mathbb{R}$ is a globally increasing function and takes as input a random vector uniformly distributed on $[0,1]^d$. Alleviating the notations, g represents now the initial numerical function transformed in a globally increasing function and $\mathbf{X} \sim \mathcal{U}([0,1]^d)$. If the probability to estimate is

$$p = \mathbb{P}(g(\mathbf{X}) \leq q),$$

without loss of generality consider that g becomes $g(.) - q$ and then $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$. Exploiting the monotonicity of g is very informative. Indeed, let $t \in \mathbb{R}$, then for all $\mathbf{x}, \mathbf{y} \in [0,1]^d$ such that $\mathbf{x} \preceq \mathbf{y}$, it comes

$$g(\mathbf{x}) \geq t \Rightarrow g(\mathbf{y}) \geq t, \tag{2.2}$$

$$g(\mathbf{y}) \leq t \Rightarrow g(\mathbf{x}) \leq t. \tag{2.3}$$

This means that if $\mathbf{y}$ leads to an undesirable event then $\mathbf{x}$ leads also to this event. This information is obtained without any new run to the numerical code.

2.3 Monotonic constraints for probability estimation

2.3.1 Monotonic reliability method (MRM)

The advantages of monotonicity for probability estimation are examined in this section. Then, the accelerated Monte Carlo method developed in [16], dedicated to probability estimation, is presented.

From (2.1) consider that $g : [0,1]^d \to \mathbb{R}$ is a globally increasing function and denote $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ with $\mathbf{X} \sim \mathcal{U}([0,1]^d)$. To alleviate the notations, denote

$$\begin{aligned} \mathbb{U}^- &:= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) \leq 0\}, \\ \mathbb{U}^- &:= \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) > 0\}, \end{aligned}$$

and the limit state

$$\Gamma := \{\mathbf{x} \in [0,1]^d,\ g(\mathbf{x}) = 0\}.$$

Moreover, assume that Γ is simply connex and $\mu(\Gamma) = 0$. Proposition 2.1 provides a property on Γ.

Proposition 2.1 *Let* $\mathbf{x} \in \Gamma$. *There is no* $\mathbf{y} \in \Gamma$ *such that* $\mathbf{y} \succ \mathbf{x}$ *or* $\mathbf{y} \prec \mathbf{x}$.

From Equations (2.2) and (2.3) if $g(\mathbf{x}) \leq 0$ then for all $\mathbf{y} \preceq \mathbf{x}, g(\mathbf{y}) \leq 0$. This means that from one evaluation of g on a point $\mathbf{x}$, the probability p can be rewritten as follow

$$\begin{aligned} p &= \int_{[0,1]^d} \mathbb{1}_{\{g(\mathbf{y}) \leq 0\}} d\mathbf{y}, \\ &= \int_{[0,1]^d} \mathbb{1}_{\{\mathbf{y} \preceq \mathbf{x}\}} d\mathbf{y} + \int_{[0,1]^d} \mathbb{1}_{\{g(\mathbf{y}) \leq 0;\ \mathbf{y} \not\preceq \mathbf{x}\}} d\mathbf{y}. \end{aligned}$$

The first term of the integral does not depends of g and a lower bound for p is obtained:

$$p \geq \int_{[0,1]^d} \mathbb{1}_{\{\mathbf{y} \preceq \mathbf{x}\}} d\mathbf{y}.$$

More generally, let $A \subset [0,1]^d$, and denote

$$\begin{aligned} \mathbb{U}^-(A) &:= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^-} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \preceq \mathbf{x}\}, \\ \mathbb{U}^+(A) &:= \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^+} \{\mathbf{u} \in [0,1]^d,\ \mathbf{u} \succeq \mathbf{x}\}, \end{aligned}$$

with $\mathbb{U}^-(\emptyset) = \{0\}^d = (0, \ldots, 0)$ and $\mathbb{U}^+(\emptyset) = \{1\}^d = (1, \ldots, 1)$. Denote μ the Lebesgue measure on $\mathbb{R}^d$. Then, the limit surface Γ and p can be surely bounded by:

$$\begin{aligned} \mathbb{U}^-(A) \subset \mathbb{U}^- \subset [0,1]^d \backslash \mathbb{U}^+, \\ \mu(\mathbb{U}^-(A)) \leq p \leq 1 - \mu(\mathbb{U}^+(A)). \end{aligned}$$

Taking into account the monotonicity property, a method provided in [16] to get two non trivial bounds for p is provided in the following paragraph.

2.3.2 Initialisation

The monotonic hypothesis allows to get two deterministic bounds for p. Indeed, Proposition 2.1 allows to say that there exists $\mathbf{u}$ and $\mathbf{v}$ in

$$\Delta := \{\mathbf{x} = (x_1, \ldots, x_d) \in [0,1]^d :\ x_1 = \cdots = x_d\},$$

such that $g(\mathbf{u}) \leq 0$ and $g(\mathbf{v}) \geq 0$. In [16], it is assumed that g takes its values in $\{-1, 1\}$. Then, it is proposed to apply a dichotomy procedure on Δ to get these bounds (see Algorithm 2.1). In a more general case, roots finding method can be also used (see [89]).

Algorithm 2.1 Bounding p

$\mathbf{x}_{old}, \mathbf{x}_{new} \leftarrow \{1/2\}^d = (1/2, \ldots, 1/2) \in [0,1]^d$
$y_{old}, y_{new} \leftarrow g(\mathbf{x}_{old})$
while $y_{new} > 0$ **do**
  $\mathbf{x}_{old} \leftarrow \mathbf{x}_{new}$
  $y_{old} \leftarrow y_{new}$
  $\mathbf{x}_{new} \leftarrow \mathbf{x}_{old}/2$
  $y_{new} \leftarrow g(\mathbf{x}_{new})$
end while
return $(\mathbf{x}_{old}, y_{old}, \mathbf{x}_{new}, y_{new})$

2.3.3 Estimating a probability by sequential uniform sampling

In this section, an accelerated Monte Carlo method, proposed in [16], devoted to probability estimation under monotonicity constraints is presented. Assume that $(\mathbf{x}^-, \mathbf{x}^+) \in \mathbb{U}^- \times \mathbb{U}^+$ has been obtained by Algorithm 2.1. Denote

$$\begin{aligned}\mathbb{U}_0^- &= \mathbb{U}^-(\mathbf{x}^-),\\ \mathbb{U}_0^+ &= \mathbb{U}^+(\mathbf{x}^+),\end{aligned}$$

and

$$\begin{aligned}p_0^- &= \mu(\mathbb{U}_0^-),\\ p_0^+ &= 1-\mu(\mathbb{U}_0^+).\end{aligned}$$

From construction, if $(\mathbf{x}, \mathbf{y}) \in \mathbb{U}_0^- \times \mathbb{U}_0^+$ then $(\mathbf{x}, \mathbf{y}) \in \mathbb{U}^- \times \mathbb{U}^+$ (see Figure 2.1a). From now denote the *non-dominated* set the subset of $[0,1]^d$ where the sign of g is unknown. At this step, the non-dominated set is denoted

$$\mathbb{U}_0 = [0,1]^d \backslash (\mathbb{U}_0^- \cup \mathbb{U}_0^+).$$

The strategy of simulation consists in simulating sequentially in this non-dominated set. The first step of the algorithm is detailed (see Figure 2.1b). At step 1, let $\mathbf{X}_1$ be uniformly distributed on $\mathbb{U}_0$, then

$$\mathbb{P}(\mathbf{X}_1 \in \mathbb{U}^-) = \frac{p - p_0^-}{p_0^+ - p_0^-},$$

and

$$\begin{aligned}\mathbb{U}_1^- &= \mathbb{U}^-(\mathbf{X}_1) \cup \mathbb{U}_0^-,\\ \mathbb{U}_1^+ &= \mathbb{U}^+(\mathbf{X}_1) \cup \mathbb{U}_0^+,\\ \mathbb{U}_1 &= [0,1]^d \backslash (\mathbb{U}_1^- \cup \mathbb{U}_1^+).\end{aligned}$$

Finally,

$$\mu(\mathbb{U}_1^-) = p_1^- \leq p \leq p_1^+ = 1 - \mu(\mathbb{U}_1^+).$$

For all $k \geq 1$, let $\mathbf{X}_k$ be uniformly distributed on the non-dominated set $\mathbb{U}_{k-1}$ and denote $\mathcal{F}_{k-1} = \sigma(\mathbf{X}_j,\ 1 \leq j \leq k)$ with $\mathcal{F}_0 = \{\emptyset, \mathcal{P}([0,1]^d)\}$. It comes

$$\begin{aligned}\mathbb{U}_k^- &= \mathbb{U}^-(\{\mathbf{X}_1, \ldots, \mathbf{X}_k\}) \cup \mathbb{U}_0^-,\\ \mathbb{U}_k^+ &= \mathbb{U}^+(\{\mathbf{X}_1, \ldots, \mathbf{X}_k\}) \cup \mathbb{U}_0^+,\\ \mathbb{U}_k &= [0,1]^d \backslash (\mathbb{U}_k^- \cup \mathbb{U}_k^+),\end{aligned}$$

and

$$\mu(\mathbb{U}_k^-) = p_k^- \leq p \leq p_k^+ = 1 - \mu(\mathbb{U}_k^+).$$

All these properties are summarised in Figure 2.1. At step n, it is proposed to estimate p as the maximiser of the likelihood function given by

$$L_n(r) = L_n(\mathbf{x}_1, \ldots, \mathbf{x}_n | r) = \prod_{k=1}^{n} \left(\frac{r - p_{k-1}^-}{p_{k-1}^+ - p_{k-1}^-}\right)^{\mathbb{1}_{\{\mathbf{x}_k \in \mathbb{U}^-\}}} \left(\frac{p_{k-1}^+ - r}{p_{k-1}^+ - p_{k-1}^-}\right)^{1-\mathbb{1}_{\{\mathbf{x}_k \in \mathbb{U}^-\}}}, \tag{2.4}$$

and p is estimated by

$$\hat{p}_n = \underset{r\in]p_{n-1}^-,p_{n-1}^+[}{\arg\max}\ L_n(r).$$

Maximizing L_n is equivalent to maximize $l_n = \log(L_n)$. Since l_n is a concave function, find the maximum of l_n is equivalent to find the root of its derivative. Denoting

$$\begin{aligned}\omega_k(r)^{-1} &= (r - p_{k-1}^-)(p_{k-1}^+ - r),\\ p_k &= p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\xi_{\mathbf{x}_k},\end{aligned}$$

the maximum of l_n is given by

$$\hat{p}_n = \frac{\displaystyle\sum_{k=1}^{n}\omega_k(\hat{p}_n)p_k}{\displaystyle\sum_{k=1}^{n}\omega_k(\hat{p}_n)}.$$

In practice, $\hat{p}_n$ is obtained by maximizing (2.4), and the Fisher information associated to $\hat{p}_n$ is defined by

$$J_n(p) = \sum_{k=1}^{n}\mathbb{E}[\omega_k(p)].$$

The following theorem [16] provides conditions to ensure that the estimator $\hat{p}_n$ is asymptotically normal.

Theorem 2.1 *Let $(\lambda_n)_{n\geq 1}$ be a deterministic sequence in $]0,1[$ such that $\lambda_n \underset{n\to+\infty}{\longrightarrow} 1$. If*

(i) $\frac{1}{n^\delta}\sum_{k=1}^{n}(\omega_k(p) - \mathbb{E}[\omega_k(p)]) \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 0$ *for all* $\delta > 0$.

(ii) $\frac{p_n^+ - p}{p - p_n^-} \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 0$.

(iii) $\frac{\bar{p}_n - p}{p_n^+ - p} \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 0$ *and* $\frac{\bar{p}_n - p}{p - p_n^-} \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 0$ *with* $\bar{p}_n = (1-\lambda_n)\hat{p}_n + \lambda_n p$,

then

$$J_n^{1/2}(p)(\hat{p}_n - p) \underset{n\to+\infty}{\xrightarrow{\mathcal{L}}} \mathcal{N}(0,1).$$

In this theorem, the quantity $J_n(p)$ is unknown, but the following proposition provides an asymptotic approximation.

Proposition 2.2 *Denote $\hat{J}_n(p) = \sum_{k=1}^{n}\omega_k(p)$. If (ii) and (iii) of Theorem 2.1 hold and if:*

(iv) hypothesis (i) holds for all $\delta \geq 1/2$.

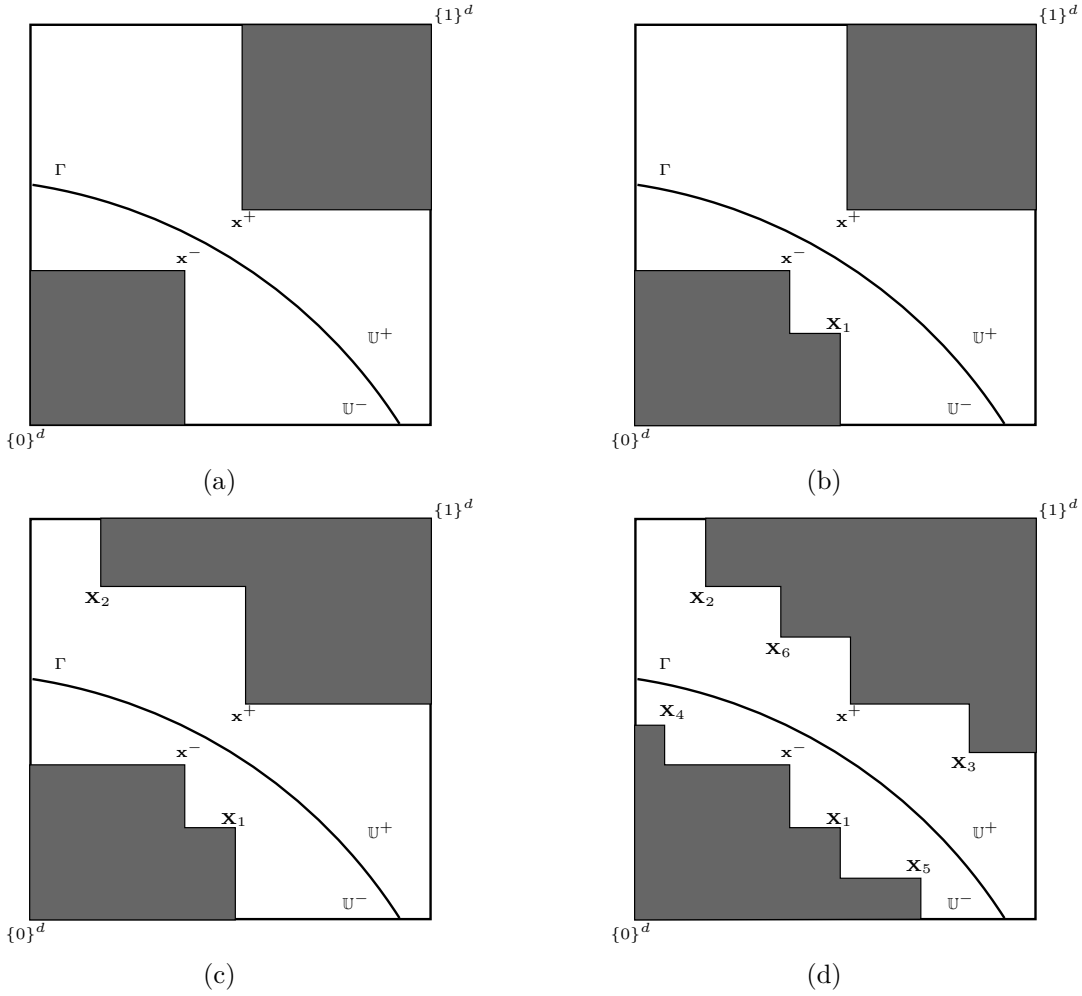


Figure 2.1: Illustration in dimension 2. **Up-Left:** the two points $\mathbf{x}^-, \mathbf{x}^+$ are provided by Algorithm 2.1. The gray squares represent $\mathbb{U}_0^-$ and $\mathbb{U}_0^+$. **Up-Right:** $\mathbf{X}_1$ is uniformly distributed on the non-dominated set $\mathbb{U}_0$ then $\mathbb{U}_0^-$ is updated. **Down-Left:** $\mathbf{X}_2$ is uniformly distributed on the non-dominated set $\mathbb{U}_1$ then $\mathbb{U}_1^+$ is updated. **Down-Right:** after six simulations, it comes $\mathbb{U}_6^- = \mathbb{U}^-(\{\mathbf{x}^-, \mathbf{X}_1, \mathbf{X}_4, \mathbf{X}_5\})$ and $\mathbb{U}_6^+ = \mathbb{U}^+(\{\mathbf{x}^+, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_6\})$.

$$\textit{(v)}\ \nexists n < \infty,\ p = \frac{1}{2n} \frac{\sum_{k=1}^{n} \omega_k(p)(p_{k-1}^+ + p_{k-1}^-)}{\sum_{k=1}^{n} \omega_k(p)},$$

then

$$\frac{\hat{J}_n(p)^{5/2}}{|\hat{J}_n'(p)|} \left(\hat{J}_n^{-1}(\hat{p}_n) - J_n^{-1}(p) \right) \xrightarrow[n \to +\infty]{\mathcal{L}} \mathcal{N}(0,1).$$

As discussed in Chapter 3, the rate of convergence of the bounds depends on the dimension. Since the variance of $\hat{p}_n$ depends on the bounds, it increases with the dimension for a given n.

2.4 Adaptation of classical methods to rare event estimation under monotonicity constraint

2.4.1 Monotone Monte Carlo

Assume that $p = \mathbb{P}(g(\mathbf{X}) \leq 0)$ with $\mathbf{X} \sim \mathcal{U}nif([0,1]^d)$. Let $(\mathbf{X}_k)_{k\geq 1}$ be an iid sequence of random vectors uniformly distributed on $[0,1]^d$. The standard Monte Carlo method provides the following estimator of p

$$\widehat{p}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{\{g(\mathbf{X}_k)\leq 0\}}.$$

Let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be n random vectors uniformly distributed on $[0,1]^d$, If at step n there exists a point $\mathbf{X}_0$ in $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^-$ (resp. $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^+$) such that $\mathbf{X}_{n+1} \preceq \mathbf{X}_0$ (resp. $\mathbf{X}_{n+1} \succeq \mathbf{X}$) then without more calls to g it is known that $\mathbf{X}_{n+1} \in \mathbb{U}^-$ (resp. $\mathbf{X}_{n+1} \in \mathbb{U}^+$). Finally, the sign of g is known on $n+1$ points but only n runs have been made. Assume there is n calls to g available, then the probability p can be estimated by

$$\frac{1}{N_n}\sum_{k=1}^{N_n} \mathbb{1}_{\{g(\mathbf{X}_k)\leq 0\}}, \tag{2.5}$$

where N_n is a random variable equals to the total number of simulations drawn until g was called n times. It must be noticed that N_n is random since at each step k, the probability to save an evaluation is equal to $1 - \mathbb{E}[p^+_{k-1} - p^-_{k-1}]$.

The difficulty to know the distribution of N_n makes difficult to have theoretical results. Such methods have been previously studied in [100].

2.4.2 Monotone Subset Simulation

As indicated in Section 1.8, the main difficulty in subset simulation is to determine the intermediate subsets. The monotonicity hypothesis is helpful to have information on these subsets. Recall that p can be rewritten as $p = \prod_{k=1}^{n} \mathbb{P}(\mathbf{X} \in F_k | \mathbf{X} \in F_{k-1})$ with $\mathbf{X} \sim \mathcal{U}nif([0,1]^d)$, $F_0 = [0,1]^d$ and for all $k \geq 1$, $F_k \subset F_{k-1}$. As said in the previous section, a non-dominated set is provided by two points given by Algorithm 2.1:

$$\begin{aligned}
\mathbb{U}_0^- &= \mathbb{U}^-(\mathbf{x}^-),\\
\mathbb{U}_0^+ &= \mathbb{U}^+(\mathbf{x}^+),\\
\mathbb{U}_0 &= [0,1]^d \backslash (\mathbb{U}_0^+ \cup \mathbb{U}_0^-).
\end{aligned}$$

Denote

$$\begin{aligned}
p_0^- &= \mu(\mathbb{U}_0^-),\\
p_0^+ &= 1 - \mu(\mathbb{U}_0^+).
\end{aligned}$$

For instance, the only subset including $\mathbb{U}^-$ is $F_0 = [0,1]^d \backslash \mathbb{U}_0^+$. Then p can be rewritten as

$$\begin{aligned}
p &= \mathbb{P}(\mathbf{X} \in \mathbb{U}^- | \mathbf{X} \in F_0)\mathbb{P}(\mathbf{X} \in F_0),\\
&= \mathbb{P}(\mathbf{X} \in \mathbb{U}^- | \mathbf{X} \in F_0)p_0^+.
\end{aligned} \tag{2.6}$$

The conditional probability $\mathbb{P}(\mathbf{X} \in \mathbb{U}^-|\mathbf{X} \in F_0)$ can be estimated by a standard Monte Carlo method. Nonetheless, Equation (2.6) does not take into account the lower bound p_0^-. Replacing F_0 by $\mathbb{U}_0$, the probability p becomes

$$p = p_0^- + (p_0^+ - p_0^-)\mathbb{P}(\mathbf{X} \in \mathbb{U}^-|\mathbf{X} \in \mathbb{U}_0),$$

and can be generalised at any step n by

$$p = p_{n-1}^- + (p_{n-1}^+ - p_{n-1}^-)\mathbb{P}(\mathbf{X} \in \mathbb{U}^-|\mathbf{X} \in \mathbb{U}_{n-1}).$$

Finally, let $(\mathbf{X}_k^{(n)})_{k\geq 1}$ be an independent sequence of random vectors uniformly distributed on the non-dominated set $\mathbb{U}_{n-1}$, then p can be estimated by

$$\hat{p}_N = p_{n-1}^- + \frac{p_{n-1}^+ - p_{n-1}^-}{N} \sum_{k=1}^{N} \mathbb{1}_{\{\mathbf{X}_k^{(n)} \in \mathbb{U}^-\}}. \tag{2.7}$$

Nevertheless, this strategy is not optimal. If one of the simulations is in the non-dominated set then the bounds cannot be updated. Moreover, estimator (2.7) does not take advantage of these new bounds contrary to the estimator provided in Section 2.3.1.

The subset simulation adapted to monotonic situations can be summarized in two mains stages. The first one consists in making a sample from any distribution and build the deterministic bounds as well as the non-dominated set. The second one is a standard Monte Carlo procedure restricted to the non-dominated set which provide the estimator (2.7).

Another approach is to make the first stage with all the simulations available, then the second stage consists in estimating the sign of the numerical code g by $\tilde{g}$. Since it is sufficient to know the sign of g, this is a problem of binary classification. The estimator becomes

$$\widetilde{p}_N = p_n^- + \frac{p_n^+ - p_n^-}{N} \sum_{k=1}^{N} \mathbb{1}_{\{\tilde{g}(\mathbf{X}_k^{(n)})\leq 0\}},$$

with $(\mathbf{X}_k^{(n)})_{k\geq 1}$ an iid sequence of random vectors uniformly distributed on the non-dominated set $\mathbb{U}_n$.

Chapter 3 provides more precise results on the rate of convergence of the bounds obtained from a Monte Carlo or a sequential framework.

2.4.3 Parallel MRM

From the beginning it is assumed that only one evaluation at a time by the numerical code is available. If the numerical code can be applied to a design of experiments, an estimator can be easily adapted from [16]. For $k \geq 1$, let $(\mathbf{X}_j^{(k)})_{j\geq 1}$ be a sequence of independent random vectors uniformly distributed on the non-dominated set $\mathbb{U}_{k-1}$. Assume at each step k the size of the sample is equal to m_k. The conditional likelihood function becomes

$$L_n(r) = \prod_{k=1}^{n} \prod_{j=1}^{m_k} \left(\frac{r - p_{k-1}^-}{p_{k-1}^+ - p_{k-1}^-}\right)^{\mathbb{1}_{\{\mathbf{X}_j^{(k)} \in \mathbb{U}^-\}}} \left(\frac{p_{k-1}^+ - r}{p_{k-1}^+ - p_{k-1}^-}\right)^{1-\mathbb{1}_{\{\mathbf{X}_j^{(k)} \in \mathbb{U}^-\}}}. \tag{2.8}$$

The following proposition provides the maximum of (2.8).

Proposition 2.3 *The conditional maximum likelihood estimator is equal to*

$$\hat{p}_n = \frac{\sum_{k=1}^{n}\sum_{j=1}^{m_k}\omega_k(\hat{p}_n)p_{k,j}}{\sum_{k=1}^{n} m_k\omega_k(\hat{p}_n)},$$

where $\omega_k(\hat{p}_n)^{-1} = (r - p_{k-1}^-)(p_{k-1}^+ - r)$ *and* $p_{k,j} = p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{\mathbf{X}_j^{(k)} \in \mathbb{U}^-\}}$.

Equivalent results than (2.1) can be obtained by replacing condition (i) by

$$\frac{1}{n^\delta}\sum_{k=1}^{n} m_k\left(\omega_k(p) - \mathbb{E}[\omega_k(p)]\right) \xrightarrow[n\to+\infty]{\mathbb{P}} 0 \text{ for all } \delta > 0.$$

Proof of Proposition 2.3. Define $l_n(r) = log(L_n(r))$, then

$$l_n(r) = \sum_{k=1}^{n}\sum_{j=1}^{N_k}\mathbb{1}_{\{\mathbf{X}_j^{(k)}\in\mathbb{U}^-\}}\log\left(\frac{r - p_{k-1}^-}{p_{k-1}^+ - p_{k-1}^-}\right) + (1 - \mathbb{1}_{\{\mathbf{X}_j^{(k)}\in\mathbb{U}^-\}})\log\left(\frac{p_{k-1}^+ - r}{p_{k-1}^+ - p_{k-1}^-}\right),$$

and its derivative is

$$\begin{aligned} l_n'(r) &= \sum_{k=1}^{n}\sum_{j=1}^{N_k}\frac{p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{\mathbf{X}_j^{(k)}\in\mathbb{U}^-\}} - r}{(r - p_{k-1}^-)(p_{k-1}^+ - r)} \\ &= \sum_{k=1}^{n}\sum_{j=1}^{N_k}\omega_k(r)(p_{k,j} - r), \end{aligned}$$

where

$$\begin{aligned} \omega_k(r)^{-1} &= (r - p_{k-1}^-)(p_{k-1}^+ - p) \\ p_{k,j} &= p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{\mathbf{X}_j^{(k)}\in\mathbb{U}^-\}}. \end{aligned}$$

The maximum likelihood estimator verifies

$$\sum_{k=1}^{n}\sum_{j=1}^{m_k}\omega_k(\hat{p}_n)(p_{k,j} - \hat{p}_n) = 0,$$

then it can be expressed as

$$\hat{p}_n = \frac{\sum_{k=1}^{n}\sum_{j=1}^{m_k}\omega_k(\hat{p}_n)p_{k,j}}{\sum_{k=1}^{n} m_k\omega_k(\hat{p}_n)}.$$

□

2.5 Empirical improvement of the deterministic bounds

In this section, an empirical method to improve the convergence of the bounds is presented. The results of this section comes from [88]. The aim is to choose a point in the non-dominated set which maximises a criterion in the aim to reduce the upper bound. They are based on the volume contribution of the upper bound. Denote for all $\mathbf{x} \in [0,1]^d$

$$\mathbb{V}^-(\mathbf{x}) := \{\mathbf{u} \in [0,1]^d, \mathbf{u} \preceq \mathbf{x}\},$$
$$\mathbb{V}^+(\mathbf{x}) := \{\mathbf{u} \in [0,1]^d, \mathbf{u} \succeq \mathbf{x}\}.$$

These notations will be used further in Chapter 5. For $n \geq 1$, these two criteria, the so-called volume-maximin (V-Maximin) and classification-maximin (C-Maximin) criteria, are respectively defined by

$$V(\mathbf{x}) = \min\left[\mu(\mathbb{U}^-(\mathbf{X}_1,\dots,\mathbf{X}_{n-1}) \cup \mathbb{V}^-(\mathbf{x})) - p^-_{n-1}, p^+_{n-1} - \mu(\mathbb{U}^+(\mathbf{X}_1,\dots,\mathbf{X}_{n-1}) \cup \mathbb{V}^+(\mathbf{x}))\right],$$
$$C(\mathbf{x}) = [p^+_{n-1} - \mu(\mathbb{U}^+(\mathbf{X}_1,\dots,\mathbf{X}_{n-1}) \cup \mathbb{V}^+(\mathbf{x}))]\pi_1(\mathbf{x}),$$

where $\pi_1(\mathbf{x})$ is the weight that $\mathbf{x}$ is in $\mathbb{U}^+$ obtained from monotonic neural networks recently developed in [111]. The criterion V represents the volume contribution of a point for the deterministic bounds (see Figure 2.2). The minimum in the definition of V can be seen as a binary classifier of Γ. Let $\mathbf{x}$ be a point in the non-dominated set $\mathbb{U}_{n-1}$. It can be reasonable to say that if its volume contribution to p^-_{n-1} is lower than its contribution to p^+_{n-1} then it is assumed that $\mathbf{x}$ is in $\mathbb{U}^-$. Finally, the aim is to maximise this volume contribution to find a point as close as possible to Γ. The criterion C represents the contribution of a point to the upper deterministic bounds according to a classification weight.

Finding the maximum of these criteria on the non-dominated set is difficult since they are not expressed in a closed form. Moreover, the computation of the contribution to the upper bound involved in these criteria is time-consuming. Then, the maximum has been chosen among a sample uniformly distributed on the non-dominated set.

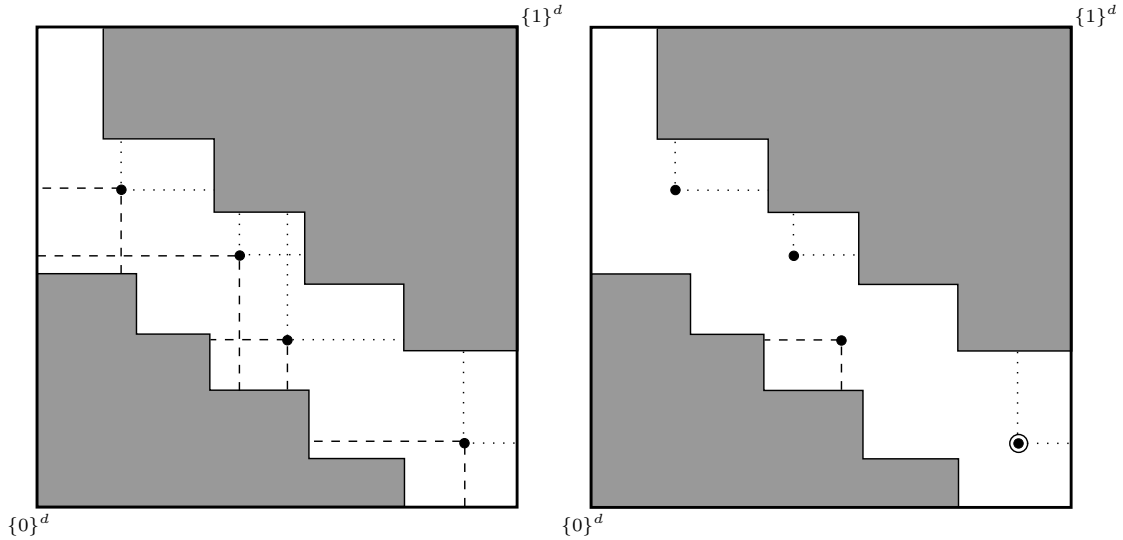


Figure 2.2: Illustration of the criterion V. The black dots represent some candidates to be the maximiser of V. **Left**: dashed and dotted lines delimit respectively the volume contribution to the lower and upper deterministic bounds. **Right**: the minimum of these contributions for each candidates is retained. The encircled point is chosen as the maximiser of V.

Criteria	Deterministic bounds	$(3, 10^{-4}, 200)$	$(5, 10^{-4}, 250)$	$(6, 10^{-3}, 300)$
Uniform	p_n^- $(\times p)$	0.44	0.14	0.70
	p_n^+ $(\times p)$	3.68	14.7	24
V-maximin	p_n^- $(\times p)$	0.43	0.02	0.02
	p_n^+ $(\times p)$	4.88	15.4	22.9
C-maximin	p_n^- $(\times p)$	0.20	0.1	0.03
	p_n^+ $(\times p)$	2.1	6.0	11.9

Table 2.1: Comparison of the influence of different criteria to reduce the upper bound of p.

The numerical example used in this comparison is now presented. Let $d \geq 2$ and $\mathbf{x} = (x^1, \ldots, x^d)$ in $[0,1]^d$, the function is defined by

$$g(\mathbf{x}) = \frac{x^1}{\sum_{i=1}^d x^i}.$$

Let $\mathbf{X} = (X^1, \ldots, X^d)$ be a random vector such that $X^i \sim \Gamma(i+1, 1)$. Then, $g(\mathbf{X}) \sim \mathcal{B}eta(2, (d+1)(d+2)/2 - 3)$. Let $q_{d,p}$ be the p-quantile of $g(\mathbf{X})$, then the probability p is defined by $p = \mathbb{P}(g(\mathbf{X}) - q_{d,p} \leq 0)$. Results obtained are summarised in Table 2.1 for different value of (d, p, n). It is also compared the value of the bounds obtained for a uniform sampling within the non-dominated set. This first exploration shows that a deterministic framework can reduce significantly one of the two deterministic bounds.

As for the FORM/SORM coupled with importance sampling, the maximiser of such a criterion can be used to calibrate an importance density. For example, a Gaussian distribution truncated on the non-dominated space, centred on such point with a small variance.

2.6 Conclusion

In this chapter, some existing methods for probability estimation have been adapted in the monotonic cases. Except for the parallel MLE construction, they do not totally exploit the information provided by the monotonic hypothesis. The numerical code studied in this thesis cannot be called in parallel then the parallel MLE is not usable in this thesis. The next chapter of this thesis aim to improve the understanding of the behaviour of these bounds. More particularly, their rate of convergence, in some sense, is studied for different strategy of sampling: a standard Monte Carlo based sampling and a sequential sampling in the non-dominated set. The obtained results will be more deeply investigated in Chapter 4 to control a probability estimator.

Chapter 3

Approximation of limit state surfaces in monotonic Monte Carlo settings

This chapter has been submitted as a research article in 2015 under the same title.

Résumé Ce chapitre étudie les propriétés théoriques de convergence d'estimateurs produits par un plan d'expérience d'une fonction monotone ayant pour entrée un vecteur. La quantité à estimer est une probabilité associée à un événement indésirable et la fonction étudiée est un code de calcul. Comme décrit dans le chapitre 2, deux bornes déterministes de cette probabilité peuvent être obtenues. Deux types de plan d'expériences sont étudiés pour étudier la vitesse de convergence de ces bornes. Le premier est construit de manière indépendante et le second séquentiellement. De plus, un estimateur consistant de la surface (ou état limite) séparant les ensembles, sous des hypothèses isotoniques et de régularités, peut être construit. Sa vitesse de convergence vers le vrai état limite peut être déterminée. Cet estimateur est construit par l'agrégation de Machine à Vecteur Support (SVM) utilisé comme classifieur binaire. Les résultats numériques obtenus sur des exemples jouets mettent en lumière que la construction est plus rapide que celle proposée par les réseaux de neurones monotones récemment développés mais avec une même qualité de prédiction.

Abstract. This chapter investigates the theoretical convergence properties of the estimators produced by a numerical exploration of a monotonic function with multivariate random inputs in a structural reliability framework. The quantity to be estimated is a probability typically associated to an undesirable (unsafe) event and the function is usually implemented as a computer model. As said in Chapter 2, two deterministic bounds of this probability can be obtained. Two frameworks have been considered to study their rate of convergence. Moreover, a consistent estimator of the (limit state) surface separating the subsets under isotonicity and regularity arguments can be built, and its convergence speed can be exhibited. This estimator is built by aggregating semi-supervized binary classifiers chosen as constrained Support Vector Machines (SVM). Numerical experiments conducted on toy examples highlight that they work faster than recently developed monotonic neural networks with comparable predictable power.

3.1 Introduction

Due to the increasingly powerful computational tools, so-called computer models g are developed to mimic complex processes (e.g., technical, biological, socio-economical, etc. [23, 37]). The

exploration of multivariate deterministic *black-box* functions $x \mapsto g(\mathbf{x})$ by numerical designs of experiments has become, over the last years, a major theme of research in the interacting fields of engineering, numerical analysis, probability and statistics [108]. Numerical investigations based on Monte Carlo variance reduction techniques [103] are often needed since, on the one hand, the complexity of g restricts the use of intrusive explorations, and in the other hand each run of the computer model can be very costly. A number of studies have for common aim to delimit the set of input situations $\mathbf{x}$ dominated by a limit state surface Γ defined by the equation $g(\mathbf{x}) = 0$, where the dominance rule is defined by a partial order. In multi-objective optimization frameworks, where g is assimilated to a decision rule, Γ can be viewed as a Pareto frontier delimiting a performance space [55]. In structural reliability it is often wanted to assess the measure of the set $\mathbb{U}^- = \{x \in \mathbb{R}^d, g(\mathbf{x}) \leq 0\}$, which can be interpreted as the probability p of an undesirable event when the input vector $\mathbf{x}$ is randomized [77].

The Pareto dominance between the inputs $\mathbf{x}$ is the natural partial order needed to formalize an assumption of monotonicity placed on g. This hypothesis corresponds to a technical reality in numerous situations encountered in engineering [38] or decision-making [14], or a conservative approximation to reality in reliability studies. Therefore the exploration of monotonic time-consuming computer models by numerical means has been investigated by a growing number of researchers [16, 33, 59]. The most immediate benefit of monotonicity is surrounding the isotonic limit state surface Γ by two Pareto-dominated sets delimited by the elements of the numerical design, the measures of which defining deterministic bounds for the probability p [38]. Recent works focused on the estimation of these bounds and the computation of statistical estimators of p based on Monte Carlo numerical designs [16].

While random designs appear as powerful tools for surrounding Γ and building estimators of p, the stochastic behaviour of the associated random sets and their measures has not been investigated yet. The aim of this chapter is to fill in this gap by establishing first convergence results for these objects from random set theory. A corollary of these results, under a convexity argument, is the consistency of an estimator of Γ built from the hull of the numerical design, accompanied with its convergence rate. Such an estimator is proposed by a combination of Support Vector Machines (SVM). It appears as a faster alternative in large dimensions to neural networks specifically developed for isotonic situations [111], while both techniques are usual in structural reliability frameworks [68].

This chapter is organized as follows. Section 3.2 details the main concepts and notations. The convergence of a dominated set is examined in Section 3.3. First we study conditions leading to convergence. Then, under some regularity assumption on the limit state, a rate of convergence for an estimator of Γ is provided in term of Hausdorff distance. Lastly, more precise results are given for two particular cases. Taking into account monotonicity properties, Section 3.4 focuses on a sequential sampling strategy. A non naive acceptance-rejection method to simulate in the so-called non-dominated set is provided . Then the convergence rates of bounds is compared numerically with a standard Monte Carlo and a sequential sampling. These theoretical results are derived in Section 3.5 to build a consistent semi-adaptive SVM-based classifier of the limit state surface Γ respecting isotonic and convexity constraints. Numerical experiments conducted in Section 3.5.2 complete this analysis by comparing the properties of the classifier with the recently developed monotonic neural networks. The proofs of the new technical results are postponed to Section 3.7.

3.2 Framework

Recall the definitions given in Chapter 2.

Let $\mathbf{X}$ be a random vector uniformly distributed on $[0,1]^d$ and g be a measurable application from $[0,1]^d$ to $\mathbb{R}$. The function g discriminates two sets of input situations, leading to undesirable and safe events, respectively. These classes are denoted $\mathbb{U}^- = \{\mathbf{x} \in [0,1]^d : g(\mathbf{x}) \leq 0\}$ and $\mathbb{U}^+ = \{\mathbf{x} \in [0,1]^d : g(\mathbf{x}) > 0\}$, with $[0,1]^d = \mathbb{U}^- \cup \mathbb{U}^+$. Hence the probability p that an undesirable event may occur is

$$p = \mathbb{P}(g(\mathbf{X}) \leq 0) = \mathbb{P}(\mathbf{X} \in \mathbb{U}^-).$$

Estimating p by a minimum number of calls to g is a prominent subject of interest in engineering. This number can be drastically diminished when g is assumed to be monotonous. Characterizing the monotonicity of g requires to use the Pareto dominance between two elements of $\mathbb{R}^d$, recalled in next definition.

Definition 3.1 *Let $\mathbf{u} = (u_1, \ldots, u_d)$ and $\mathbf{v} = (v_1, \ldots, v_d) \in \mathbb{R}^d$. We say that $\mathbf{u}$ is (resp. strictly) dominated by $\mathbf{v}$, denoting $\mathbf{u} \preceq \mathbf{v}$ (resp. $\mathbf{u} \prec \mathbf{v}$), if for all $i = 1, \ldots, d$, $u_i \leq v_i$ (resp. $u_i < v_i$).*

Assumption 3.1 *Let $g : [0,1]^d \subset \mathbb{R}^d \to \mathbb{R}$. It is assumed that g is globally increasing, i.e. for all $\mathbf{u}, \mathbf{v} \in [0,1]^d$ such that $\mathbf{u} \preceq \mathbf{v}$, then $g(\mathbf{u}) \leq g(\mathbf{v})$.*

Remark 3.1 *As pointed in Chapter 2 and [16], any monotonic function can be reparametrized to be increasing with respect to all its inputs.*

This property has for consequence to simplify the exploration of $\mathbb{U}^-$ and $\mathbb{U}^+$ and provide deterministic bounds on p. Let $\mathbf{x} \in \mathbb{U}^-$ (*resp.* $\mathbb{U}^+$) and $\mathbf{y} \in \mathbb{U}$. Since g is increasing, if $\mathbf{y} \preceq \mathbf{x}$ (*resp.* $\mathbf{y} \succeq \mathbf{x}$) then $\mathbf{y} \in \mathbb{U}^-$ (*resp.* $\mathbb{U}^+$). More generally, consider a set A of elements on $[0,1]^d$ (for instance a design of numerical experiments) and define

$$\mathbb{U}^-(A) = \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^-} \{\mathbf{u} \in [0,1]^d : \ \mathbf{u} \preceq \mathbf{x}\}, \tag{3.1}$$

$$\mathbb{U}^+(A) = \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^+} \{\mathbf{u} \in [0,1]^d : \ \mathbf{u} \succeq \mathbf{x}\}. \tag{3.2}$$

Then the monotonicity property implies the following result, since $\mathbb{U}^-(A) \subset \mathbb{U}^- \subset [0,1]^d \backslash \mathbb{U}^+(A)$.

Proposition 3.1 *Denote μ the Lebesgue measure on $\mathbb{U}$. For any countable set A,*

$$\mu(\mathbb{U}^-(A)) \leq p \leq 1 - \mu(\mathbb{U}^+(A)).$$

Remark 3.2 *In general, the set A will be $\{X_1, \ldots, X_n\}$, the sets $\mathbb{U}^-(A)$ and $\mathbb{U}^+(A)$ would then be random sets and the inequality becomes an almost sure inequality.*

The *non-dominated* subset $\mathbb{U}(A) = [0,1]^d \backslash (\mathbb{U}^-(A) \cup \mathbb{U}^+(A))$ contains necessarily Γ and is the only subset of $[0,1]^d$ for which a deeper numerical exploration is needed in view of estimating Γ and p. Formally

$$\Gamma = \left(\overline{\mathbb{U}}^- \backslash \mathring{\mathbb{U}}^-\right) \cap \left(\overline{\mathbb{U}}^+ \backslash \mathring{\mathbb{U}}^+\right)$$

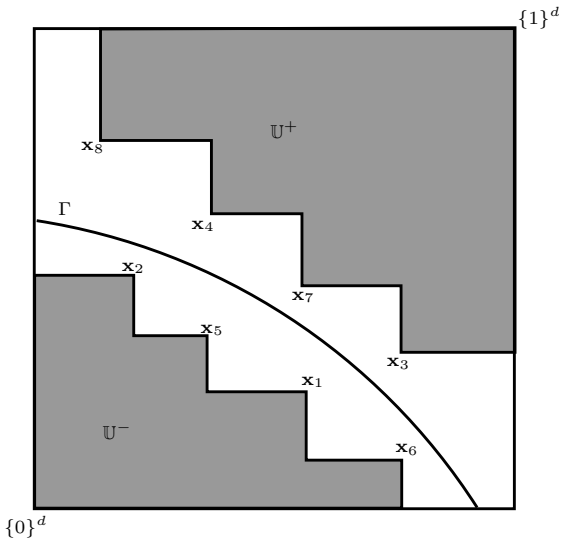


Figure 3.1: Illustration for $d = 2$ of the designs $\mathbb{U}^-(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_6)$ and $\mathbb{U}^+(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_7, \mathbf{x}_8)$.

where $\overline{E}$ and $\mathring{E}$ are respectively the closure and the interior of a set E. Next (mild) assumption is required to view the problem of estimating Γ as a binary classification problem with perfectly separable classes, and is needed to study the convergence of the dominated sets to $\mathbb{U}^-$ and $\mathbb{U}^+$ when the number of elements of the design increases. A two-dimensional illustration is provided on Figure 3.1.

Assumption 3.2 *Assume that* $\mu(\Gamma) = 0$ *and* Γ *is simply connex.*

In the remainder of the chapter, a design of numerical experiments $\{\mathbf{X}_i, g(\mathbf{X}_i)\}_{i=1,\dots,n}$ is considered. To alleviate the notations denote for all $n \geq 1$,

$$\mathbb{U}_n^- = \mathbb{U}^-\{\mathbf{X}_1, \dots, \mathbf{X}_n\},$$
$$\mathbb{U}_n^+ = \mathbb{U}^+\{\mathbf{X}_1, \dots, \mathbf{X}_n\},$$

and the non-dominated set $\mathbb{U}_n = [0,1]^d \backslash (\mathbb{U}_n^- \cup \mathbb{U}_n^+)$. The Lebesgue measures (or hypervolumes) of sets $(\mathbb{U}_n^-, \mathbb{U}_n^+)$ are then denoted p_n^- and $1 - p_n^+$, such that, from Proposition 3.1, we have with probability one

$$p_n^- \leq p \leq p_n^+. \tag{3.3}$$

These two bounds can be computed using a sweepline algorithm described in [16] at an exponential cost. A faster approximation method is provided in Section 3.4. Two situations are further investigated from the point of view of the convergence of dominated sets and bounds. First, the design is chosen static in $[0,1]^d$ (usual Monte Carlo sampling). Then a sequential Monte Carlo approach is explored, assuming $\mathbb{U}_0 = [0,1]^d$ and $\mathbf{X}_i$ being sampled within the current non-dominated set $\mathbb{U}_{i-1}$ for all $i > 0$. The convergence of the sets $\mathbb{U}_n^-$ and $\mathbb{U}_n^+$ will be studying via the Hausdorff distance defined as follows.

Definition 3.2 *Let* $\|\cdot\|_q$ *be the* $\mathcal{L}^q$ *norm on* $\mathbb{R}^d$*, that is for* $0 < q < +\infty$, $\|\mathbf{x}\|_q = (\sum_{i=1}^d |x_i|^q)^{1/q}$*, and for* $q = +\infty$, $\|\mathbf{x}\|_\infty = \max_{i=1,\dots,d_{i=1,\dots,d}} x_i$*. Let* (A, B) *be two non-empty subsets of the normed*

vector space $([0,1]^d, \|\cdot\|_q)$. *The Hausdorff distance* $d_{H,q}$ *is defined by*

$$d_{H,q}(A,B) = \max(\sup_{\mathbf{y}\in A}\inf_{\mathbf{x}\in B}\|\mathbf{x}-\mathbf{y}\|_q;\ \sup_{\mathbf{x}\in B}\inf_{\mathbf{y}\in A}\|\mathbf{x}-\mathbf{y}\|_q).$$

It is always finite since $(A,B) \subset \left([0,1]^d\right)^2$ are bounded. The usual form of the Hausdorff distance is defined for $q=2$. To alleviate notations, it is now denoted $d_H(.,.) = d_{H,2}(.,.)$ and $\|.\| = \|.\|_2$.

3.3 Convergence results

3.3.1 Almost sure convergence for general sample strategy

The sequences $(p_n^-)_n$ and $(p_n^+)_n$ are respectively increasing bounded from above and decreasing bounded from below. Hence, there exists two random variables p_∞^- and p_∞^+ such that almost surely $p_n^- \longrightarrow p_\infty^-$, $p_n^+ \longrightarrow p_\infty^+$ and $p_\infty^- \leq p \leq p_\infty^+$. In this section we provide general conditions on the design of experiments that ensures that $p_\infty^- = p_\infty^+ = p$.

Proposition 3.2 *[Independant Sampling] Let* $(\mathbf{X}_k)_{k\geq 1}$ *be a sequence of independent random vectors on* $[0,1]^d$. *If there exists* $\varepsilon_1 > 0$ *such that for all* $\mathbf{x} \in \Gamma^{\varepsilon_1} = \{\mathbf{u} \in [0,1]^d,\ d(\mathbf{u},\Gamma) < \varepsilon_1\}$ *there exists* $\varepsilon_2 > 0$ *such that*

$$\sum_{n\geq 1} \mathbb{P}(\mathbf{X}_n \in B(\mathbf{x},\varepsilon_2)) = +\infty,$$

then $(p_n^-, p_n^+) \underset{n\to+\infty}{\overset{a.s.}{\to}} (p,p)$.

Corollary 3.1 *Under Assumption 3.2 and the conditions of Proposition 3.2, then*

$$\left(d_H(\mathbb{U}_n^-, \mathbb{U}^-),\ d_H(\mathbb{U}_n^+, \mathbb{U}^+)\right) \underset{n\to+\infty}{\xrightarrow{a.s.}} (0,0).$$

Example 3.1 *If a sequence* $(X_n)_{n\in\mathbb{N}^*}$ *that is i.i.d. and uniformly distributed on* $[0,1]^d$ *then it satisfies the assumption of Proposition 3.2. More generally any sequence of iid random variables with bounded from below densities on* $[0,1]^d$ *satisfies the assumption of Proposition 3.2.*

Since the only useful experiments are those which fall into the non-dominated set $\mathbb{U}_n$, a sequential Monte Carlo strategy can be carried out, as detailed in next proposition.

Proposition 3.3 *[Sequential sampling] Let* $(\mathbf{Y}_n)_{n\geq 1}$ *be a sequence of iid random variables uniformly distributed on* $[0,1]^d$. *Let* $(T_n)_{n\geq 1}$ *be a sequence of random variables on* $\mathbb{N}$ *defined by* $T_1 = 1$ *and for all* $n \geq 1$

$$T_{n+1} = \min\{j > T_n,\ \mathbf{Y}_j \in \mathbb{U}_{T_n}\}.$$

Let $(\mathbf{X}_n)_{n\geq 1}$ *be the sequence of random variables defined for all* $n \geq 1$ *by* $\mathbf{X}_n = \mathbf{Y}_{T_n}$. *Then, conditionally to* $\mathbf{X}_1, \ldots, \mathbf{X}_{n-1}$, $\mathbf{X}_n \sim \mathcal{U}(\mathbb{U}_{T_{n-1}})$, *and* $(p_n^-, p_n^+) \overset{a.s.}{\to} (p,p)$, *where* (p_n^-, p_n^+) *are obtained from the sequence* $(\mathbf{X}_n)_{n\geq 1}$.

It is now natural to quantify the rate of convergence. While it seems difficult to provide an explicit rate in the general case. We will then provide some explicit rate in particular cases.

3.3.2 Rate of convergence under some regularity assumptions

In this section convergence in Haussdorf distance sense of $(\mathbb{U}_n^-)_n$ towards $\mathbb{U}^-$ is proven, and under regularity assumption for the set $\mathbb{U}^-$ a rate of convergence is provided. The regularity considered here is the (α, γ)-regularity introduced in [4].

Definition 3.3 *A set K is said (α, γ)-regular if there exist $\alpha, \gamma > 0$ such that for all $0 < \varepsilon \leq \gamma$ and for all $\mathbf{x} \in K$ one has*

$$\mu(\mathbf{B}(\mathbf{x}, \varepsilon) \cap K) \geq \alpha \mu(\mathbf{B}(\mathbf{x}, \varepsilon))$$

where $\mathbf{B}(\mathbf{x}, r)$ is the open ball with center $\mathbf{x}$ and radius r.

This notion was introduced in [4] to prove minimax rates in classification problems. It has also been considered in [22, 32] in order to state convergence rate when estimating sets. In particular Proposition 3.4 below can be seen as an application of Theorem 2 in [22].

Roughly speaking, this condition excludes pathological sets $\mathbb{U}^-$, as those having infinitely many sharper and sharper peaks or presenting a jagged border Γ. For instance, it is indirectly proved in [48] that if $\mathbb{U}^-$ is convex and has non-empty interior then it is regular. In a structural reliability framework this condition is conservative since it traduces the hypothesis that any combination of inputs located between two situations leading to an undesirable event leads also to such an event. Described in [112], so-called $r-$convex sets that generalize the notion of convexity are also regular [22]. It is traduced in $\mathbb{U}^-$ by the assumption (or "rolling condition", cf. [112]) that for each boundary point $\mathbf{x} \in \Gamma$, there exists $\mathbf{y} \in \mathbb{U}^-$ such that $\mathbf{x} \in B(\mathbf{y}, r)$. In practice the assumption of $r-$convexity appears mild, as small values of r can produce a large variety of limit state surfaces presenting irregularities [22].

Proposition 3.4 *Let $(\mathbf{X}_k)_{k\geq 1}$ be a sequence of iid random variables uniformly distributed on $[0,1]^d$ and $(\tilde{\mathbf{X}}_k)_{k\geq 1}$ be a sequence of iid random variables uniformly distributed on $\mathbb{U}^-$. Denote $\tilde{\mathbb{U}}_n^- = \mathbb{U}^-(\tilde{\mathbf{X}}_1, \ldots, \tilde{\mathbf{X}}_n)$. Let $(F_n)_{n\geq 1}$ be a sequence of measurable subsets of $[0,1]^d$ such that for all $n \geq 1, \mathbb{U}_n^- \subset F_n \subset [0,1]^d \backslash \mathbb{U}_n^+$. Then*

(1) *$d_H(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{a.s.} 0$ and $\mu(F_n) \xrightarrow[n\to+\infty]{a.s.} p$.*

(2) *If $\mathbb{U}^-$ is regular, then almost surely*

$$d_H(\tilde{\mathbb{U}}_n^-, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

(3) *Furthermore, if $\mathbb{U}^+$ is also regular, and if g is continuous, then almost surely*

$$d_H(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right). \tag{3.4}$$

Remark 3.3 *Note, that if in* **(2)**, *we replace $(\tilde{\mathbb{U}}_n^-)$ by $(\mathbb{U}_n^-)$, the a.s. bound becomes*

$$O\left((\log N_1/N_1)^{1/d}\right),$$

where N_1 is the number of points X_i such that $g(X_i)) < 0$ and follows the binomial distribution with parameter n and p. Moreover the continuity assumption in **(3)** *can be weakened, it is enough (see the proof) that one can cut $[0,1]^d$ following $g^{-1}(\{0\})$ and glue it.*

3.3.3 Convergence results in dimension 1

In dimension 1, we provide a complete description of the rate of convergence, in particular we prove that $n(p-p_n^-)$ and $n(p_n^+-p)$ are asymptotically exponentially distributed.

Proposition 3.5 *Let $(X_n)_{n\geq 1}$ be an iid sequence of random variables uniformly distributed on $[0,1]$ and $p\in[0,1]$. Define $p_n^- = \max_{i=1,\dots,n}(X_i\cdot \mathbb{1}_{\{X_i\leq p\}})$ and $p_n^+ = \mathbb{1}_{\{\max_{i=1,\dots,n}(X_i)\leq p\}} + \min_{i=1,\dots,n}(X_i\cdot\mathbb{1}_{\{X_i>p\}})$. Then*

$$\begin{aligned} n(p-p_n^-) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1),\\ n(p_n^+-p) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1),\\ \mathbb{E}[p_n^+-p_n^-] &= \frac{2}{n+1}-\frac{1}{n+1}\left(p^{n+1}+(1-p)^{n+1}\right), \end{aligned} \tag{3.5}$$

where $\mathcal{E}xp(\lambda)$ is the exponential distribution with density $f_\lambda(x)=\lambda\exp(-\lambda x)\mathbb{1}_{\{x\geq 0\}}$.

The cost of adopting a naive sequential strategy to decrease the volume of the non-dominated set, by sampling a new design element within $[0,1]^d$, can be appreciated by coming back to the unidimensional case ($d=1$). In this framework, denote $W_n = \min\{j\geq 1 :\ X_{n+j}\in]p_n^-,p_n^+[\}$ where $(X_{n+j})_{j\geq 1}$ is an iid sequence of random variables uniformly sampled on $[0,1]$. For all $r\geq 1$, one has

$$\begin{aligned} \mathbb{P}(W_n=r) &= \mathbb{P}(X_{n+1}\notin]p_n^-,p_n^+[,\dots,X_{n+r-1}\notin]p_n^-,p_n^+[,X_{n+r}\in]p_n^-,p_n^+[),\\ &= \mathbb{E}[\mathbb{P}(X_{n+1}\notin]p_n^-,p_n^+[,\dots,X_{n+r-1}\notin]p_n^-,p_n^+[,X_{n+r}\in]p_n^-,p_n^+[|X_1,\dots,X_n)],\\ &= \mathbb{E}[(1-(p_n^+-p_n^-))^{r-1}(p_n^+-p_n^-)]. \end{aligned}$$

From linearity of the expectation and Jensen inequality, it comes

$$\begin{aligned} \mathbb{E}[W_n] &= \mathbb{E}[(p_n^+-p_n^-)^{-1}],\\ &\geq \mathbb{E}[p_n^+-p_n^-]^{-1},\\ &\geq \frac{n+1}{2} \quad \text{from (3.5)} \end{aligned} \tag{3.6}$$

which is (as expected) a prohibitive cost for large values of n.

More accurate results can be only obtained in some particular case. The following proposition provides a result of the expected Lebesgue measure of the non-dominated set for $d=1$.

Proposition 3.6 *Let $p_0^-=0$, $p_0^+=1$, $X_1\sim\mathcal{U}([p_0^-,p_0^+])$, $\xi_1=\mathbb{1}_{\{X_1\leq p\}}$ and $\mathcal{F}_1=\sigma\{X_1\}$. For all $n\geq 1$ define conditionally to $\mathcal{F}_n=\sigma\{X_k,1\leq k\leq n\}$:*

$$\begin{aligned} X_{n+1} &\sim \mathcal{U}([p_n^-,p_n^+]),\\ \xi_{n+1} &= \mathbb{1}_{\{X_{n+1}\leq p\}}\\ p_{n+1}^- &= p_n^- + (X_{n+1}-p_n^-)\xi_{n+1},\\ p_{n+1}^+ &= p_n^+ - (p_n^+-X_{n+1})(1-\xi_{n+1}). \end{aligned}$$

Then, for $n\geq 1$,

$$\frac{1}{2^n}\leq \mathbb{E}[p_n^+-p_n^-]\leq \left(\frac{3}{4}\right)^n.$$

Corollary 3.2 *If* $p \in \{0,1\}$, $\mathbb{E}[p_n^+ - p_n^-] = 2^{-n}$.

Remark 3.4 *The shrinking convergence rate, which is inversely linear in* n *in a static Monte Carlo strategy, is significantly improved by becoming exponential when opting for a sequential strategy.*

3.3.4 Asymptotic results when $\Gamma = \{1\}^d$

Equivalent results are difficult to obtain for greater dimension and for a general limit state Γ. The followings proposition provides asymptotic results in the particulare case $\Gamma = \{1\}^d$.

Proposition 3.7 *Assume* $\Gamma = \{1\}^d$. *Let* $(\mathbf{X}_k)_{k\geq 1}$ *be a sequence of iid random variables uniformly distributed on* $[0,1]^d$. *For* $0 < q < +\infty$ *denote* $A(1,q) = 1$ *and for* $d \geq 2$,

$$A_{d,q} = \frac{1}{dq^{d-1}} \prod_{i=1}^{d-1} B(i/q, 1/q),$$

with $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$. *For all* $n \geq 1$, *let* $\mathbb{U}_n^- = \mathbb{U}^-(\mathbf{X}_1, \ldots, \mathbf{X}_n)$.

(1) *If* $0 < q < +\infty$ *then*

$$(A_{d,q}n)^{1/d} d_{H,q}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d).$$

(2) *If* $q = +\infty$ *then*

$$n^{1/d} d_{H,\infty}\left(\mathbb{U}_n^-, [0,1]^d\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d),$$

where $\mathcal{W}(1,d)$ *is the Weibull distribution with scale parameter* 1 *and shape parameter* d *having cumulative density function* $F(t) = 1 - e^{-t^d}$ *for all* $t \geq 0$.

Proposition 3.8 *Under the assumptions and notations of Proposition 3.7 we have*

$$\mathbb{E}[\mu([0,1]^d \backslash \mathbb{U}_n^-)] \underset{n\to+\infty}{\sim} \frac{\log(n)^{d-1}}{n(d-1)!}.$$

Remark 3.5 *Denoting* $C_\infty(K)$ *the convex hull of a set* K, *the result of Proposition 3.8 is tantamount to the following result provided in [5]:*

$$\mathbb{E}[\mu([0,1]^d \backslash C_\infty(\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}))] \underset{n\to+\infty}{\sim} \frac{\log(n)^{d-1}}{n}.$$

When $d = 1$, *the convergence order obtained in Proposition 3.5 is retreived.*

3.4 Sequential sampling

Using a naive methods to simulate in a constrained space can be time-consuming. Indeed, Equation 3.6 gives the expected number of simulation needed to simulate in the non-dominated set in dimension 1. A method to simulate in the non-dominated set is provided in the first part of this section. In a second part, the measure of the non-dominated set is compared in function of the strategies of simulation adopted: standard Monte Carlo or sequential sampling.

3.4.1 Fast uniform sampling within the non-dominated set

Simulating uniformly on $[0,1]^d$ can be time-consuming when the aim is to simulate in the non-dominated set. The strategy provided in this section involves to build a subset of $[0,1]^d$ containing the non-dominated set then to simulate uniformly within this subset. After n calls to g let $\mathbf{x}$ in $\mathbb{U}_n^+$. From construction, the non-dominated set is in $[0,1]^d \backslash \mathbb{U}_n^+(\mathbf{x})$, it remains to simulate uniformly on this subset of $[0,1]^d$. The choice of $\mathbf{x}$ is described further in this section. The proposed method use the following lemma coming from a simple acceptance-rejection method.

Lemma 3.1 *Let A be a measurable subset of $[0,1]^d$ with $\mu(A) > 0$ and such that $A = A_1 \cup \cdots \cup A_m$ with $A_i \cap A_j = \emptyset$ if $i \neq j$. Let $(\mathbf{X}_k)_{k\geq 1}$ be an iid sequence of random vectors satisfying for all i, $\mathbb{P}(\mathbf{X}_k \in A_i) = \mu(A_i)\backslash\mu(A)$. Let C be a measurable subset of A, and $T = \inf\{k \geq 1, \mathbf{X}_k \in C\}$ then $\mathbf{X}_T$ is uniformly distributed on C.*

The construction involves to split up $[0,1]^d$ in hyper-rectangles (see Figure 3.2) which are defined for all $i = 0, \ldots, 2^d - 1$ by

$$Q_i(\mathbf{x}) = I_i^1 \times I_i^2 \times \cdots \times I_i^d,$$

with

$$I_i^j = \begin{cases} [x^j, 1] & \text{if } b_j^i = 0, \\ [0, x^j] & \text{if } b_j^i = 1, \end{cases}$$

where $b^i = (b_1^i, \ldots, b_d^i)$ is equal to i coded in base 2. Since 0 is coded as $00\cdots0$ in base 2 with d numbers, one deduce that $Q_0(\mathbf{x}) = [x^1, 1] \times [x^2, 1] \times \cdots \times [x^d, 1] = \mathbb{U}^+(\mathbf{x})$. Let $(\mathbf{X}_k)_{k\geq 1}$ be an iid sequence of random vectors satisfying for all $i = 1, \ldots, 2^d - 1$

$$\mathbb{P}(\mathbf{X} \in Q_i(\mathbf{x})) = \frac{\mu(Q_i(\mathbf{x}))}{1 - \mu(Q_0(\mathbf{x}))}. \tag{3.7}$$

Define $T = \inf\{k \geq 1, \mathbf{X}_k \in \mathbb{U}_n\}$.Applying Lemma 3.1 with $A = Q_1(\mathbf{x}) \cup \cdots \cup Q_{2^d-1}(\mathbf{x})$ and C the non-dominated set $\mathbb{U}_n$, one deduce that $\mathbf{X}_T$ is uniformly distributed on the non-dominated set. The point $\mathbf{x}$ is chosen in order to maximise the quantity in Equation (3.7). It must be noticed that more p is close to 0 more the points of Ξ_n^+ are close to $\{0\}^d$ then the probability to have a simulation in the non-dominated set increases.

3.4.2 Suboptimal sequential framework

As said in the previous paragraph, the rate of convergence of the bounds is difficult to obtain in a sequential framework. In the following paragraphs, two sequential frameworks are described. The first one is built especially for the case $d = 2$ and the second one is a generalisation.

First framework for $d = 2$.

A first framework of sampling is now described for $d = 2$ and $\Gamma = \{\mathbf{1}\}$. Let $\mathbf{X}_1$ be uniformly distributed on $[0,1]^2$ then $p_1^- = \mu(\mathbb{U}^-(\mathbf{X}_1))$. Following Figure 3.3, the set $\mathbb{U}_1 = [0,1]^2 \backslash \mathbb{U}^-(\mathbf{X}_1)$ is split in two disjoints hyper-rectangles $E_{2,1}$ and $E_{2,2}$. Let $\mathbf{X}_{2,1}, \mathbf{X}_{2,2}$ be uniformly distributed on each of them. Each of these two random vectors allows to split $E_{2,1}$ and $E_{2,2}$ in two hyper-rectangles (see Figure 3.3 up right). Repeating on n steps, the lower bound p_n^- is defined by

$$p_n^- = \mu(\mathbb{U}^-(\mathbf{X}_1)) + \sum_{k=2}^{n} \sum_{j=1}^{2^k} \mu(\mathbb{U}^-(\mathbf{X}_{k,j}) \cap E_{k,j}),$$

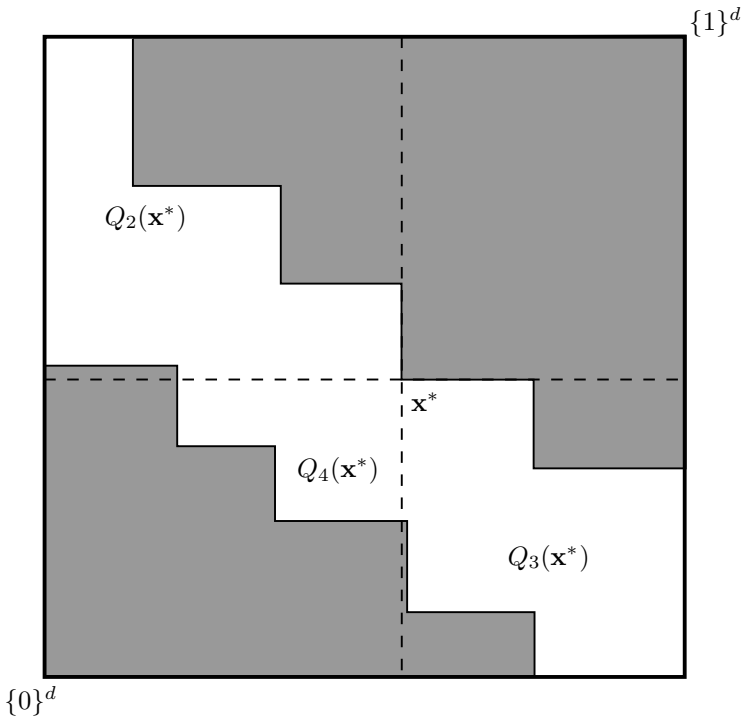


Figure 3.2: Splitting up $[0,1]^d$ from $\mathbf{x}^*$ in dimension 2. The hyper-rectangle $Q_0(\mathbf{x}^*)$ is in $\mathbb{U}^+$.

where $\mathbf{X}_{k,j}$ is uniformly distributed on $E_{k,j}$. The following proposition provides the expectation of p_n^-.

Proposition 3.9 *Let $d = 2$ and $\Gamma = \{1\}^2$. Let $k \geq 1$ be the step of the framework of simulation. Then $n = 2^k - 1$ random vectors have been simulated and*

$$\mathbb{E}[p_n^-] = 1 - \frac{1}{(n+1)^{\log(4/3)/\log 2}}.$$

Second framework for $d \geq 2$.

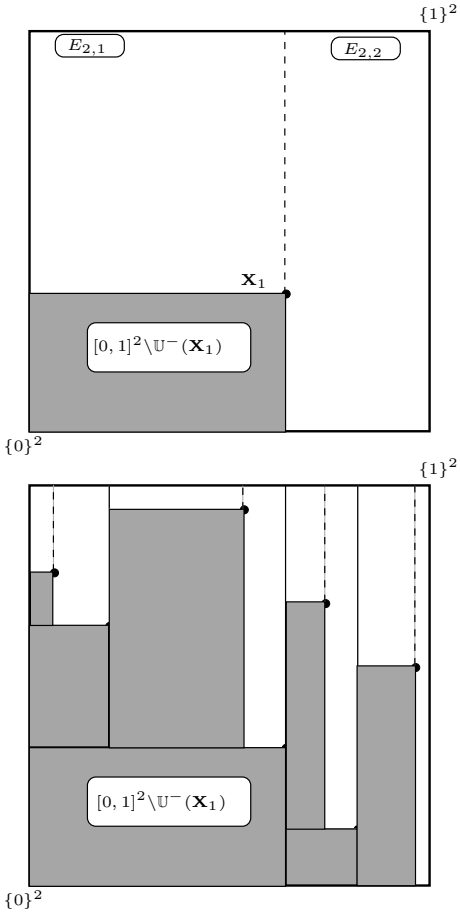
Instead to split in two subsets the non-dominated set, it is now split in $2^d - 1$ disjoints hyper-rectangles as described in Figure 3.4. In this case, the lower bound is defined as follow

$$p_n^- = \mu(\mathbb{U}^-(\mathbf{X}_1)) + \sum_{k=2}^{n} \sum_{j=1}^{(2^d-1)^k} \mu(\mathbb{U}^-(\mathbf{X}_{k,j}) \cap E_{k,j}).$$

The expectation of p_n^- is given in the proposition below.

Proposition 3.10 *Let $d \geq 2$ and $\Gamma = \{1\}^d$. Let $k \geq 1$ be the step of the framework of simulation. Then $n = \frac{(2^d-1)^k-1}{2^d-2}$ random vectors have been simulated and*

$$\mathbb{E}[p_n^-] = 1 - \frac{1}{((2^d-2)n+1)^{\log(2-2^{-d})/\log(2^d-1)}}.$$



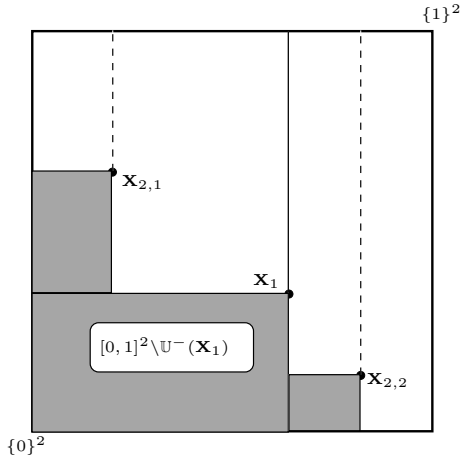


Figure 3.3: Illustration of the framework of simulation in dimension 2. The Lebesgue measure of the set in gray represents the value of p_n^-. **Up-left**: First step: a random vector is uniformly distributed on $[0,1]^d$. The dashed lines represents the frontier of the two subsets $E_{2,1}$ and $E_{2,2}$. **Up-right**: Second step: two random vectors are uniformly distributed on $E_{2,1}$ and $E_{2,2}$. These points split $E_{2,1}$ and $E_{2,2}$ in two subsets. **Down**: Third step of the framework.

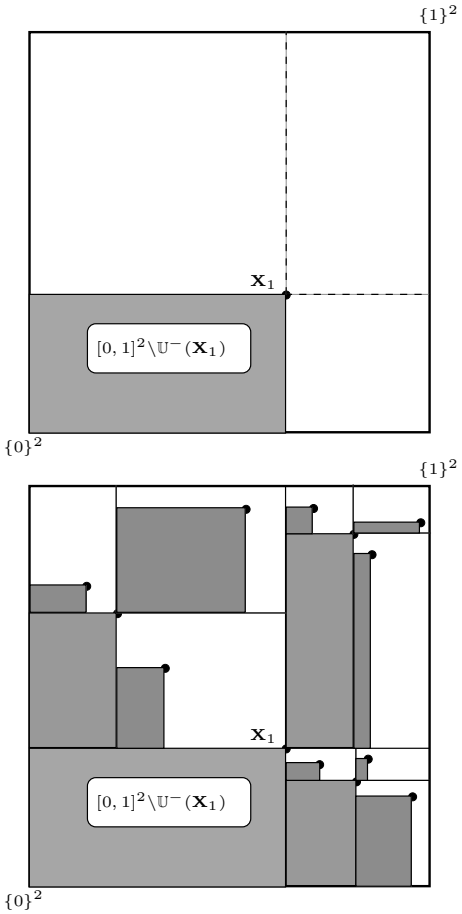


Figure 3.4: Illustration of the framework of simulation in dimension 2. The Lebesgue measure of the set in gray represents the value of p_n^-. **Up-left**: First step: a random vector is uniformly distributed on $[0,1]^d$. The dashed lines represents the frontier of the two subsets $E_{2,1}, E_{2,2}$ and $E_{2,3}$. **Up-right**: Second step: a random vector uniformly distributed on $E_{2,1}, E_{2,2}$ and $E_{2,3}$. Each of these sets are split in three subsets. **Down**: Third step of the framework.

The rates of convergence provided in Propositions 3.9 and 3.10 are greater than the one provided in Proposition 3.8. Since a sequential framework must accelerate the convergence of the lower bound, these results state that these constructions are suboptimal compared to a sequential scheme.

3.4.3 Numerical study

To complete the previous theoretical studies, a brief numerical comparison of convergence rates obtained for some static and sequential strategies is conducted in this paragraph. The impact of design strategies on the mean measure of the non-dominated set $\mathbb{U}_n$ is examined through two repeated experiments conducted over the following model, defined such that $\mathbb{U}^- = [0,1]^d$, $p = 1$ and $\Gamma = \{1\}^d$ for various dimensions $d \in \{1, 2, 3, 4, 5, 10\}$. That means, after n simulations, the volume of

$$\bigcup_{\mathbf{x} \in \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}} \{\mathbf{u} \in [0,1]^d : \ \mathbf{u} \preceq \mathbf{x}\},$$

is compared, where $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are respectively obtained from a uniform and a sequential sampling strategy. The model is run $n = 100$ times per experiment and repeated 100 times. The results of these experiments are summarized on Table 3.1. As expected, the remaining volume is lower with a sequential Monte Carlo than a standard Monte Carlo. Nonetheless, this difference decreases when the dimension increases and the remaining volume is nearly equal to 1 in dimension 10 for the two Monte Carlo simulations.

Dimension	1	2	3	4	5	10
Monte Carlo strategy	9.9×10^{-3}	0.052	0.145	0.28	0.43	0.937
Sequential strategy	7.89×10^{-31}	4.17×10^{-5}	0.017	0.14	0.33	0.935

Table 3.1: Mean measures (volumes) of non-dominated set $\mathbb{U}_n$ after $n = 100$ runs.

Remark 3.6 *Table 3.1 shows that the gain provided by a sequential sampling strategy decrease as the dimension increase. For $d = 10$, the remaining volume obtain with the two strategies of simulations are close.*

3.5 Application : estimating Γ using Support Vector Machines (SVM)

The previous sections provide convergence results for different strategies of simulations without the estimation of the limit state Γ. This section provide an estimator of the sign of g under some convexity assumption on $\mathbb{U}^-$. The proposed estimator is in fact a classifier based on linear SVMs. It will then be compared on a toy example, with the monotonic neural networks recently developed in [111].

3.5.1 Theoretical study

Dominated sets $(\mathbb{U}_n^-, \mathbb{U}_n^+)$ surrounding Γ can be improved by sampling within the non-dominated set $\mathbb{U}_n$. In view of improving a naive Monte Carlo approach to estimate p, as in [16], usual techniques like importance sampling or subset simulation should aim at targeting input situations close to Γ [15]. Such approaches can be guided by a consistent estimation of Γ, under the form of a supervised binary classification rule calibrated from $(\mathbb{U}_n^-, \mathbb{U}_n^+)$. This classifier has to agree with the following (isotonic) ordinal property of the limit state surface Γ.

Proposition 3.11 *Under Assumption 3.2 For all $\mathbf{u}, \mathbf{v} \in \Gamma$ such that $\mathbf{u} \neq \mathbf{v}$, $\mathbf{u}$ is not strictly dominated by $\mathbf{v}$.*

Respecting this constraint a monotonic neural network classifier was recently proposed in [111] and applied in [16] to structural reliability frameworks. While consistent, its computational cost remains high or even prohibitive when the size of the design $\mathbf{X}_1, \ldots, \mathbf{X}_n$ defining $(\mathbb{U}_n^-, \mathbb{U}_n^+)$ increases. Benefiting from a clear geometric interpretation, Support Vector Machines (SVM) offer an alternative to neural networks by their robustness to the curse of dimensionality [68]. A semi adaptive solution can be build from a combination of SVM when $\mathbb{U}^-$ is convex. Conversely, it can be easily adapted when $\mathbb{U}^+$ is a convex set.

Assuming $\mathbb{U}^-$ is convex, any points $\mathbf{x}$ of $\mathbb{U}^+$ can be separated from $\mathbb{U}^-$ by a hyperplane $h_{\mathbf{x}}(\mathbf{u}) = \alpha + \boldsymbol{\beta}_{\mathbf{x}}^T \mathbf{u}$ (see [101] Theorem 11.5) that maximises the minimal distance of $h_{\mathbf{x}}$ to $\mathbf{x}$ and $\mathbb{U}^-$. It is also possible to construct h satisfying the ordinal property on Assumption 3.1 (see Appendix 3.8.2 for more details).

Given the numerical experiment $D_n = (\mathbf{X}_i, y_i)_{1 \leq i \leq n} \in \mathbb{U} \times \{-1, 1\}\}$ where $y_i = 1$ if $g(\mathbf{X}_i) > 0$ and -1 otherwise. Let $\Xi_n^+ = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^+$ and $\Xi_n^- = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}^-$ and for any $\mathbf{x} \in \Xi_n^+$ define $h_{\mathbf{x}}$ as the hyperplane separating $\mathbf{x}$ from Ξ_n^-. The proposed classifier f_n is defined

by

$$f_n : [0,1]^d \to \{-1,+1\}$$
$$\mathbf{y} \mapsto \begin{cases} -1 & \text{if for all } \mathbf{X} \in \Xi_n^+,\ h_{\mathbf{X}}(\mathbf{y}) \leq 0 \\ +1 & \text{otherwise.} \end{cases}$$

Denote

$$F_n = \{\mathbf{x} \in [0,1]^d,\ f_n(\mathbf{x}) = -1\}$$

the set of all inputs classified as leading to undesirable events by f_n.

Theorem 3.1 *Assume $\mathbb{U}^-$ is convex, then*

(1) *f_n is globally increasing.*

(2) *For all $\mathbf{X} \in \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, $sign(g(\mathbf{X})) = f_n(\mathbf{X})$.*

(3) *The set F_n is a convex polyhedron.*

(4) *Furthermore if $(\mathbf{X}_k)_{k\geq 1}$ is a sequence of independent random vectors uniformly distributed on $[0,1]^d$, then*

$$d_H(F_n, \mathbb{U}^-) \xrightarrow[n\to+\infty]{a.s.} 0,$$

and almost surely,

$$d_H(F_n, \mathbb{U}^-) = O\left((\log n/n)^{1/d}\right).$$

Updating the classifier given a new design element $\mathbf{X}_{n+1}$ found in $\mathbb{U}^+$ can be done by a partially (semi) adaptive principle, illustrated on Figure 3.5 in dimension 2. To get f_{n+1} it is enough to build the hyperplane $h_{\mathbf{X}_{n+1}}$ which separates $\mathbf{X}_{n+1}$ from $\Xi_{n+1}^- = \Xi_n^-$ (unfortunately, if $\mathbf{X}_{n+1} \in \mathbb{U}^-$ all hyperplanes must be rebuild, but this occurs rarely with low probability $p - p_n^-$ when the design is sampled uniformly on $[0,1]^d$). If $\mathbf{X}_{n+1} \not\preceq \mathbf{x}$ for all $\mathbf{x} \in \Xi_n^-$, the support vectors and the hyperplanes will not be modified.

3.5.2 Numerical experiments

This section examines first the gain of including the constraint of monotonicity in the calibration of SVM (detailed in Appendix 3.8.2), with respect to usual SVM and the constrained neural networks proposed in [111] when the associated code is globally monotonic.

First, the case of a linear limit state is studied. Next, the comparison will be made on a function which verifies the monotonicity and convexity properties. The rate of good classification obtained with the monotonic classifier and the monotonic neural networks are compared at each step of the algorithm.

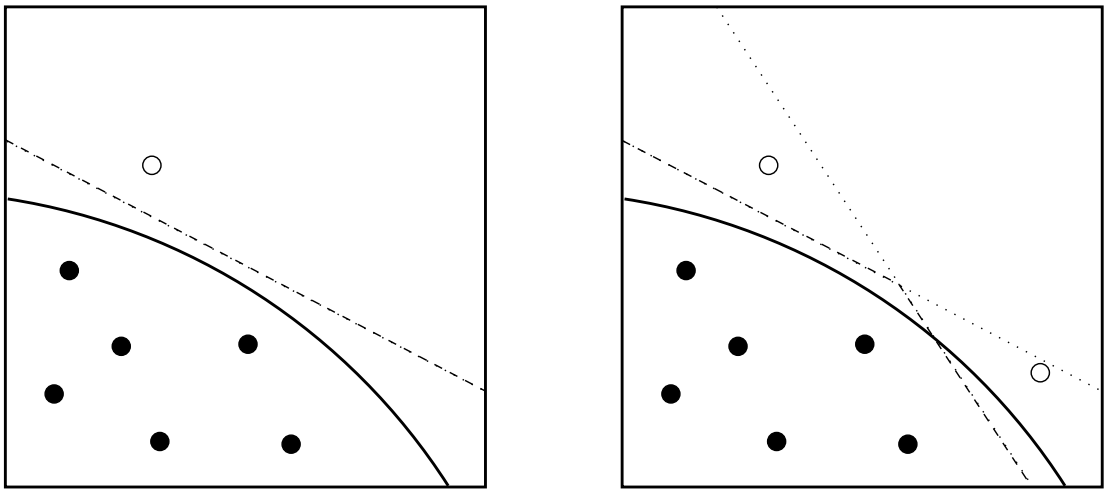


Figure 3.5: Construction of the classifier based on SVM. The plain line represent Γ and black (resp. white) points are in $\mathbb{U}^-$ (resp. $\mathbb{U}^+$). The dotted lines represents $\{\mathbf{x} : h_{\mathbf{X}}(\mathbf{x}) = 0\}$ for some $\mathbf{X}$ in $\mathbb{U}^+$. **Left**: there is one point in $\mathbb{U}^+$ then the dashed line is both $\{\mathbf{x} : h_{\mathbf{X}}(\mathbf{x}) = 0\}$ and the frontier of the classifier. **Right**: Dotted lines represent the two sets $\{\mathbf{x} : h_{\mathbf{X}}(\mathbf{x}) = 0\}$ for $\mathbf{X}$ represented by the two white points. Dashed line represent the frontier of the classifier.

Linear case.

Investigating the efficiency of the new classifier, let us start the numerical experiments with the linear case. The function g is a hyperplane defined by

$$h(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \mathbf{x},$$

with positive parameters which are, in dimension d, uniformly distributed on the unit sphere

$$\mathbb{S}_d^+ = \{\mathbf{x} \in \mathbb{R}_+^d : \ \|\mathbf{x}\| = 1\}.$$

Various sample size N are used to examine the rate of good classification when the monotonicity is taking into account or not. For each ordered pair (d, N), a hyperplane is build with parameters uniformly distributed on $\mathbb{S}_d^+$. Then, 200 data are simulated on the unit sphere to be predicted. The rate of good classification is compared for linear SVM and linear monotonic SVM. That step is repeated 100 times for each hyperplane, and also repeated with 100 differents hyperplane.

The results are summarised by Table 3.2 where there is respectively a proportion 0.1, 0.2, 0.3, 0.4 and 0.5 among N which are in $\{\mathbf{x} : \ g(\mathbf{x}) \leq 0\}$. In general, taking account of the monotonicity provides better results. As expected, more N is great and d is low more the rate of good classification increase for the two methods. Results are comparable for $d = 2$ but for a fixed N the difference grows when d increase. Finally, the constrained SVM provides significantly better results for small N.

Convex case.

In this section section, the classifier given in Theorem 3.1 is tested on a toy example. Let $\mathbf{U} = (U^1 \dots, U^d)$ be uniformly distributed on $[0, 1]^d$, and denote

$$Z_d = (U^1 U^2 \cdots U^d)^2.$$

	$d=2$	$d=10$	$d=20$	$d=40$	$d=50$	$d=100$
$N=10$	78.24/**78.40**	58.14/**62.72**	53.05/**57.65**	50.80/**53.53**	50.49/**52.32**	50.09/**50.88**
$N=20$	84.04/**84.19**	63.77/**68.99**	56.66/**62.20**	52.37/**56.83**	51.80/**55.85**	50.40/**52.83**
$N=40$	91.22/**91.34**	71.65/**74.13**	62.64/**67.38**	56.25/**61.44**	54.59/**59.91**	51.68/**55.75**
$N=50$	93.04/**93.07**	73.92/**76.41**	65.04/**70.05**	57.43/**63.04**	56.16/**61.65**	52.36/**56.87**
$N=75$	92.57/**92.74**	78.72/**80.34**	68.95/**72.59**	60.24/**66.02**	58.71/**63.99**	53.73/**58.58**
$N=100$	94.41/94.41	83.36/**84.43**	73.66/**76.95**	64.50/**69.03**	62.11/**67.26**	55.54/**60.96**
$N=200$	97.09/97.09	88.87/**89.30**	82.09/**83.55**	72.83/**76.07**	69.55/**73.22**	61.24/**66.50**
$N=10$	84.86/**85.73**	62.79/**68.05**	57.78/**63.67**	53.08/**58.46**	52.75/**57.92**	50.95/**54.41**
$N=20$	90.41/**90.44**	70.27/**74.69**	61.98/**68.49**	57.64/**63.62**	55.88/**62.30**	52.56/**57.87**
$N=40$	93.65/**93.67**	78.89/**80.90**	70.51/**74.88**	62.48/**68.38**	60.23/**66.30**	55.44/**61.93**
$N=50$	95.51/95.51	82.06/**83.78**	72.30/**76.28**	64.84/**70.50**	62.62/**68.35**	57.01/**63.39**
$N=75$	96.28/**96.29**	85.36/**86.99**	77.63/**80.25**	69.12/**73.65**	66.41/**71.86**	59.88/**66.15**
$N=100$	96.87/96.87	88.44/**89.18**	81.04/**82.94**	72.39/**76.21**	69.71/**74.30**	62.02/**67.80**
$N=200$	98.48/**98.49**	93.37/**93.63**	88.37/**89.19**	81.10/**83.18**	78.02/**80.40**	69.65/**74.13**
$N=10$	86.55/**87.42**	67.76/**73.2**	60.98/**68.26**	56.49/**63.74**	55.67/**62.95**	52.86/**59.30**
$N=20$	92.39/**92.68**	74.94/**78.36**	67.68/**73.32**	61.13/**68.43**	58.87/**66.38**	55.40/**62.69**
$N=40$	95.27/**95.40**	83.20/**85**	75.22/**78.75**	66.73/**72.70**	65.15/**71.56**	59.24/**66.67**
$N=50$	95.35/**95.37**	85.12/**86.46**	77.81/**81.04**	69.39/**74.81**	66.72/**72.89**	61.00/**68.37**
$N=75$	97.06/97.06	88.88/**89.42**	82.24/**84.51**	73.64/**78.11**	71.40/**76.19**	63.80/**70.19**
$N=100$	97.90/97.90	91.55/**91.79**	85.68/**86.71**	77.47/**80.77**	74.69/**78.78**	66.59/**72.88**
$N=200$	98.73/98.73	95.16/**95.26**	91.53/**91.94**	85.40/**86.99**	82.95/**85.09**	74.62/**79.08**
$N=10$	88.53/**89.14**	68.99/**74.31**	64.20/**71.35**	59.36/**67.83**	58.06/**66.78**	55.23/**64.17**
$N=20$	92.10/**92.22**	77.54/**81.02**	70.60/**75.69**	63.92/**71.54**	62.88/**70.56**	58.04/**66.88**
$N=40$	95.25/**95.26**	85.06/**86.52**	77.75/**81.31**	70.62/**76.28**	68.48/**74.74**	62.33/**70.43**
$N=50$	97.20/**97.20**	87.02/**87.74**	80.91/**83.52**	72.58/**77.60**	70.60/**75.88**	63.98/**71.50**
$N=75$	97.51/**97.51**	90.99/**91.54**	84.84/**86.84**	76.90/**80.92**	74.86/**79.32**	67.58/**74.37**
$N=100$	97.98/**97.99**	92.86/**93.06**	87.65/**88.57**	80.11/**83.03**	77.88/**81.74**	70.57/**76.39**
$N=200$	98.81/**98.81**	95.95/**96.07**	93.50/**93.83**	88.27/**89.44**	86.55/**88.18**	78.6/**82.36**
$N=10$	88.91/**89.52**	70.12/**75.03**	64.23/**71.39**	59.93/**68.41**	59.17/**67.97**	56.88/**66.48**
$N=20$	93.44/**93.59**	77.65/**80.22**	71.63/**76.71**	65.03/**72.34**	63.77/**71.59**	59.74/**68.46**
$N=40$	95.85/95.85	86.33/**87.23**	78.95/**81.88**	71.58/**77.11**	69.16/**75.35**	64.00/**71.87**
$N=50$	96.28/**96.36**	88.25/**89.00**	81.36/**84.56**	74.22/**79.23**	71.72/**77.56**	65.78/**73.53**
$N=75$	97.51/**97.51**	91.01/**91.61**	85.96/**87.60**	78.60/**82.51**	75.85/**80.09**	69.66/**76.18**
$N=100$	98.04/**98.04**	93.18/**93.46**	89.03/**89.81**	81.92/**84.43**	79.55/**82.73**	72.08/**78.06**
$N=200$	99.07/**99.08**	96.18/**96.27**	93.57/**94.12**	89.05/**90.12**	87.16/**88.70**	80.09/**83.76**

Table 3.2: Rate of good classification for usual SVM (left) and monotonic SVM (right) in function of d and N. From up to down, there is respectively a proportion of 0.1, 0.2, 0.3, 0.4, 0.5 in the set $\{\mathbf{x} : \ g(\mathbf{x}) \leq 0\}$.

Let $q_{d,p}$ be the p-quantile of Z_d and define the function $g(\mathbf{U}) = Z_d - q_{d,p}$. This quantile can be deduced from Equation (3.15) in Section 3.7. Indeed, let $t \in [0,1]$, then

$$\mathbb{P}(Z_d \leq t) = \mathbb{P}(U^1 U^2 \cdots U^d \leq \sqrt{t}) = \int_0^{\sqrt{t}} \frac{(-\log(x))^{d-1}}{(d-1)!} dx.$$

The function g is globally increasing and the set $\{\mathbf{u} \in [0,1]^d, g(\mathbf{u}) > 0\}$ is convex. The SVM-based classifier and the constrained neural networks are built from the points of a sequential design used to delimit the non-dominated set. At step n, the number of points which delimits this non-dominated set is denoted N_n.

The comparison is conducted as follows. At each step $n \geq 1$, $M = 500$ random vectors are uniformly distributed on the non-dominated set $\mathbb{U}_{n-1}$. The rate of good classification on these M random vectors and the time needed to build the two classifiers are stored. Then non-dominated set is updated with a random vector uniformly distributed on it. The comparison is conducted for different dimensions: 2, 3 , 4 and 5 and for $p = 0.01$ and have been averaged on 40 independent experiments.

In Figure 3.6 the rate of good classification is compared in function of N_n. The red values represent the number of times the situation where the non-dominated set is delimited by N_n points. For some $n \geq 1$, this number is greater than 40. Indeed, the sequence $(N_n)_{n\geq 1}$ is not monotonic: at any step n, a new simulation employed to update the non-dominated set $\mathbb{U}_{n-1}$ can dominate one of the point of its frontier. Then the value of N_n can be lower than N_{n-1}. At step n, having $N_n = n$ means no points of the sequential design is dominated by one another. The results show that the mean rate of good classification is equivalent for the two classifier and for the tested dimension. But, when the dimension increases the SVM-based classifier is more stable than the constrained neural networks.

Denote $(T_n^{SVM})_{n\geq 1}$ and $(T_n^{NN})_{n\geq 1}$ respectively the cumulative time needed to build the SVM-based classifier and the constrained neural networks until step n. The ratio T_n^{NN}/T_n^{SVM} of time needed to construct these estimator are compared on Figure 3.7. It is shown in this example that the semi-adaptive SVM-based classifier is less time-consuming than the non-adaptive constrained neural networks for different dimensions. Nonetheless, when the dimension increases the gain of time decrease.

3.6 Conclusion

This chapter initiates a theoretical counterpart to the increasing developments linked to the exploration of computer models that make use of their geometrical properties. The original framework is related to structural reliability problems, but the obtained results can be adapted to multi-objective optimisation contexts, characterized by strong constraints of monotonicity. The main results presented here are the consistency and law convergence of the main ingredients (random sets, deterministic bounds, classification tools) used to solve such problems in the common case where the numerical design is generated by a Monte Carlo approach. Therefore such results should be understood as benchmark objectives for more elaborated approaches. For example, use the monotonic estimation of the limit state surface proposed in this chapter which presents good theoretical and practical properties.

These more elaborated approaches are, obviously, based on sequential sampling within the current non-dominated set and will be investigated in Chapter 4.

Monotonicity of computer models can be exploited to get deterministic bounds of output quantiles. A preliminary result is given beneath in dimension 1, in corollary of Proposition 3.5,

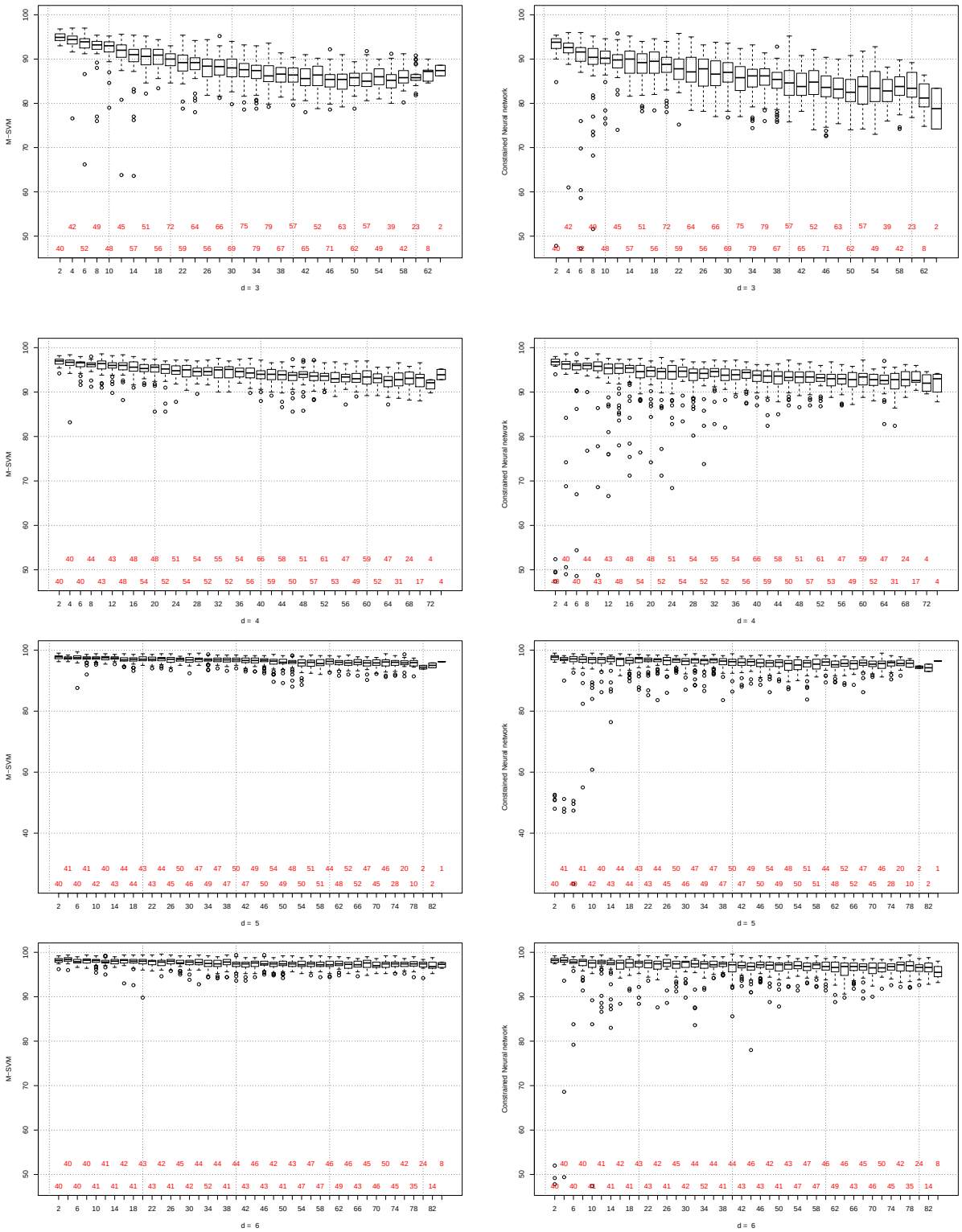


Figure 3.6: Boxplots of the rate of good classification for $d \in \{3, 4, 5, 6\}$, in function of n. In red are the sample size used for each boxplot. Left: Monotonic SVM. Right: Constrained neural networks. The results have been averaged on 40 independent experiments.

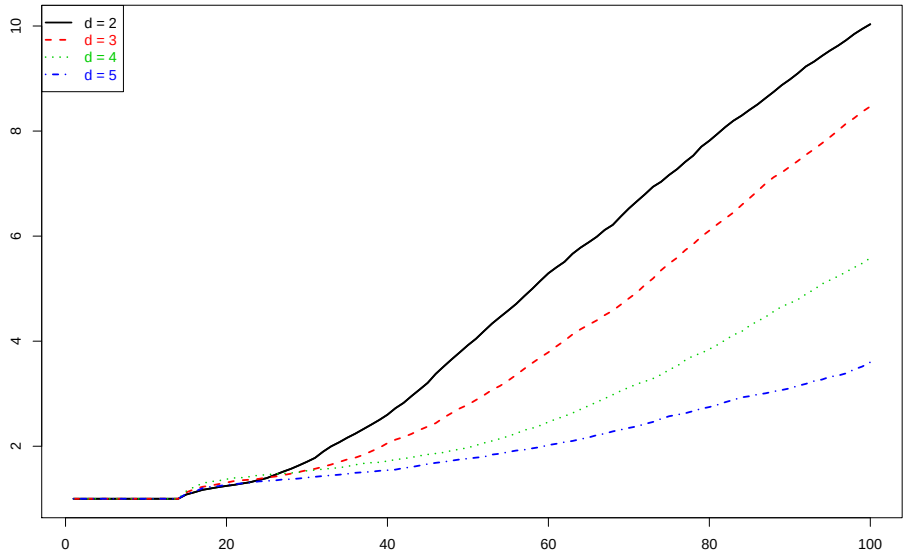


Figure 3.7: Ratio of time construction of the SVM-based classifier and the constrained neural network in function of n. The results have been averaged on 40 independent experiments.

to introduce this theme, which will be investigated in Chapter 5.

Corollary 3.3 *Assume that g is continuous, strictly increasing and differentiable at p. Denote $q_p = \inf_{q\in\mathbb{R}} (\mathbb{P}(g(\mathbf{X}) \leq q) = p)$ the p-order quantile of $g(\mathbf{X})$ and g' the derivative of g. Then*

$$n(q_p - g(p_n^-)) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)),$$
$$n(g(p_n^+) - q_p) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)).$$

3.7 Proofs

Proof of Proposition 3.2. Set $\mathbb{U}_\infty^- = \bigcup_{k\geq 1} \mathbb{U}_k^-$, $\mathbb{U}_\infty^+ = \bigcup_{k\geq 1} \mathbb{U}_k^+$ and the non-dominated set $\mathbb{U}_\infty = [0,1]^d \setminus (\mathbb{U}_\infty^- \cup \mathbb{U}_\infty^+)$. We define $p_\infty^- = \mu(\mathbb{U}_\infty^-)$ and $p_\infty^+ = \mu([0,1]^d \setminus \mathbb{U}_\infty^+)$. By inclusion and closure, the bounded sequences $(p_n^-)_{n\geq 1}$ and $(p_n^+)_{n\geq 1}$ are respectively increasing and decreasing, then $p_n^- \overset{a.s.}{\to} p_\infty^-$ and $p_n^+ \overset{a.s.}{\to} p_\infty^+$. Assume $\mathbb{U}_\infty$ is a non-empty open set such that $\mu(\mathbb{U}_\infty) = p_\infty^+ - p_\infty^- > 0$. There exist $\varepsilon_1 > 0$ and $\mathbf{x}_0 \in [0,1]^d$ such that

$$\mathbf{B}(\mathbf{x}_0, \varepsilon_1) \quad \subset \quad \mathbb{U}_\infty. \tag{3.8}$$

Hence no element of the sequence $(X_k)_{k\geq 1}$ belongs to $\mathbf{B}(\mathbf{x}_0, \varepsilon_1)$. Now, we introduce the events $A_n = \{\mathbf{X}_n \in \mathbf{B}(\mathbf{x}_0, \varepsilon_1)\}$, the are independent by construction and

$$\sum_{n\geq 1} \mathbb{P}(A_N) = \sum_{n\geq 1} \frac{\pi^{d/2} \varepsilon^d}{\Gamma(d/2+1)} = +\infty.$$

We can then apply to Borel-Cantelli's lemma that ensures that $\mathbb{P}(\limsup_n A_n) = 1$. Therefore it exists almost surely at least one $X_k \in \mathbf{B}(\mathbf{x}_0, \varepsilon_1)$, which is contradictory with (3.8). Hence $\mu(\mathbb{U}_\infty) = 0$ and necessarily $p_\infty^- = p_\infty^+ = p$ almost surely. □

Proof of Corollary 3.1. From construction, one has $\mathbb{U}_n^- \subset \mathbb{U}^-$ then from Proposition 3.2

$$d_H(\mathbb{U}_n^-, \mathbb{U}^-) = \sup_{\mathbf{x}\in\mathbb{U}_n^-} \inf_{\mathbf{y}\in\mathbb{U}^-} \|\mathbf{x}-\mathbf{y}\| \leq \mu(\mathbb{U}^-) - \mu(\mathbb{U}_n^-) = p - p_n^-, \underset{n\to+\infty}{\longrightarrow} 0.$$

□

Proof of Proposition 3.3. It is an alternative proof to the consistency result given in [16]. For any measurable set $A \subset [0,1]^d$. If $(Y_n)_n$ is an i.i.d. sequence of variable uniformly distributed on $[0,1]^d$ and $T = \inf\{n, X_n \in A\}$. Then, it is a well known fact that Y_T is uniformly distributed on A. Hence conditionnaly to Conditionally to $\mathbf{X}_1, \ldots, \mathbf{X}_{n-1}$ one has

$$\mathbf{X}_n \sim \mathcal{U}(\mathbb{U}_{T_{n-1}}).$$

Denote $(q_n^-)_{n\geq 1}$ and $(q_n^+)_{n\geq 1}$ the sequences of bounds obtained from $(\mathbf{Y}_n)_{n\geq 1}$. By construction, the sequences $(p_n^-)_{n\geq 1}$ and $(p_n^+)_{n\geq 1}$ are subsequences of $(q_n^-)_{n\geq 1}$ and $(q_n^+)_{n\geq 1}$. Then

$$p_n^+ - p_n^- \leq q_n^+ - q_n^- \underset{n\to+\infty}{\overset{a.s.}{\to}} 0.$$

□

Proof of Proposition 3.4. Proof of **(1)**. By triangle inequality, one has

$$\begin{aligned} d_H(F_n, \mathbb{U}^-) &\leq d_H(F_n, \mathbb{U}_n^-) + d_H(\mathbb{U}_n^-, \mathbb{U}^-), \\ &\leq d_H([0,1]^d \backslash \mathbb{U}_n^+, \mathbb{U}_n^-) + d_H(\mathbb{U}_n^-, \mathbb{U}^-), \text{ since } \mathbb{U}_n^- \subset F_n \subset [0,1]^d \backslash \mathbb{U}_n^+ \\ &\leq d_H([0,1]^d \backslash \mathbb{U}_n^+, \mathbb{U}^-) + 2d_H(\mathbb{U}_n^-, \mathbb{U}^-), \text{ by a second triangle inequality.} \end{aligned}$$

We know from Corollary 3.1, that

$$\begin{aligned} d_H([0,1]^d \backslash \mathbb{U}_n^+, \mathbb{U}_n^-) &\underset{n\to+\infty}{\overset{a.s.}{\to}} 0, \\ d_H(\mathbb{U}_n^-, \mathbb{U}^-) &\underset{n\to+\infty}{\overset{a.s.}{\to}} 0, \end{aligned}$$

hence $d_H(F_n, \mathbb{U}^-) \underset{n\to+\infty}{\overset{a.s.}{\to}} 0$. Besides $\mu(\mathbb{U}_n^-) \leq \mu(F_n) \leq \mu([0,1]^d \backslash \mathbb{U}_n^+)$ by construction, then $p_n^- \leq \mu(F_n) \leq p_n^+$, and from Corollary 3.1, it can be deduced that

$$\mu(F_n) \underset{n\to+\infty}{\overset{a.s.}{\to}} p,$$

that concludes the proof of **(1)**. The proof of **(2)** and **(3)** are based on the following Theorem due to [22].

Theorem 3.2 *([22], Theorem 2). Let K be a compact set on $\mathbb{R}^d$ and standard with respect to a measure ν. Let $(\mathbf{X}_k)_{k\geq 1}$ be a sequence of i.i.d. random variables uniformly distributed on K, and K_n be a set such that for all n large enough one has almost surely $(\mathbf{X}_1, \ldots, \mathbf{X}_n) \subset K_n \subset K$. Then almost surely*

$$d_H(K_n, K) = O\left(\left(\frac{\log n}{n}\right)^{1/d}\right).$$

The notion of standard set generalizes the notion of (α, γ)-regularity that is used here. Hence any (α, γ)- regular set is also standard in the sense of Theorem 3.2.
Proof of **(2)**. It is a direct consequence of Theorem 3.2.
Proof of **(3)**. Application of Theorem 3.2. In order to do so, we will cut $[0,1]^d$ following $g^{-1}(\{0\})$ and glue it on $[0,1]^{d-1}$ in the following way. Set

$$G : [0,1]^{d-1} \longrightarrow [0,1]$$
$$x \mapsto \begin{cases} \inf\{t \in [0,1], g(x,t) = 0\} & \text{if} \quad \{t \in [0,1], g(x,t) = 0\} \neq \emptyset, \\ 0 & \text{if} \quad \{t \in [0,1], g(x,t) = 0\} = \emptyset, \end{cases}$$

and define $\tilde{\Gamma}$ as the graph of the application $x \mapsto G(x) - 1$. Now let

$$\Psi : [0,1]^d \longrightarrow \mathbb{R}^d$$
$$(x,y) \mapsto \begin{cases} (x,y) & \text{if} \quad g(x,y) \leq 0, \\ (x, y-1) & \text{if} \quad g(x,y) > 0. \end{cases}.$$

We define now $\tilde{\mathbb{U}} = \Psi([0,1]^d) \bigcup \tilde{\Gamma}$ (we have cut the space $[0,1]^d$ following $g^{-1}(\{0\})$ and glue together $[0,1]^{d-1} \times \{0\}$ and $[0,1]^{d-1} \times \{1\}$, see Figure 3.8). Now by assumption $\tilde{\mathbb{U}}$ is a compact set and the random variables $(\Psi(X_i))_i$ are i.i.d uniformly distributed in $\tilde{\mathbb{U}}$. The result follows applying Theorem 3.2. □

Proof of Proposition 3.5. Let $x \in [0,p]$. Then

$$\begin{aligned} \mathbb{P}(p_n^- \leq x) &= \mathbb{P}(\max_{i=1,\dots,n} (X_i \mathbb{1}_{\{X_i \leq p\}}) \leq x) = \mathbb{P}(X_1 \mathbb{1}_{\{X_1 \leq p\}} \leq x)^n \\ &= \Big(1 - \mathbb{P}(X_1 \mathbb{1}_{\{X_1 \leq p\}} > x)\Big)^n = (1 - \mathbb{P}(p \geq X_1 > x))^n = (1 - p + x)^n. \end{aligned} \tag{3.9}$$

Then for all $x \in [0,1]$,

$$\mathbb{P}(p_n^- \leq x) = \begin{cases} 0 & \text{if } x < 0, \\ (1-p+x)^n & \text{if } 0 \leq x \leq p, \\ 1 & \text{if } x > p. \end{cases} \tag{3.10}$$

Let $x > 0$ and n be large enough such that $x \in [0, np]$, then

$$\mathbb{P}(n(p - p_n^-) < x) = 1 - (1 - x/n)^n \xrightarrow[n \to +\infty]{} (1 - e^{-x}),$$

hence $n(p - p_n^-) \xrightarrow[n \to +\infty]{\mathcal{L}} \mathcal{E}xp(1)$. From (3.10), it comes:

$$\mathbb{E}[p_n^-] = \int_0^1 \mathbb{P}(p_n^- \geq x) dx = p - \frac{1}{n+1}[1 - (1-p)^{n+1}].$$

Denote $p_n^-(p)$ a random variable such that for all $x \in [0,1]$,

$$\mathbb{P}(p_n^-(p) \leq x) = \begin{cases} 0 & \text{if } x < 0, \\ (1-p+x)^n & \text{if } 0 \leq x \leq p, . \\ 1 & \text{if } x > p. \end{cases}$$

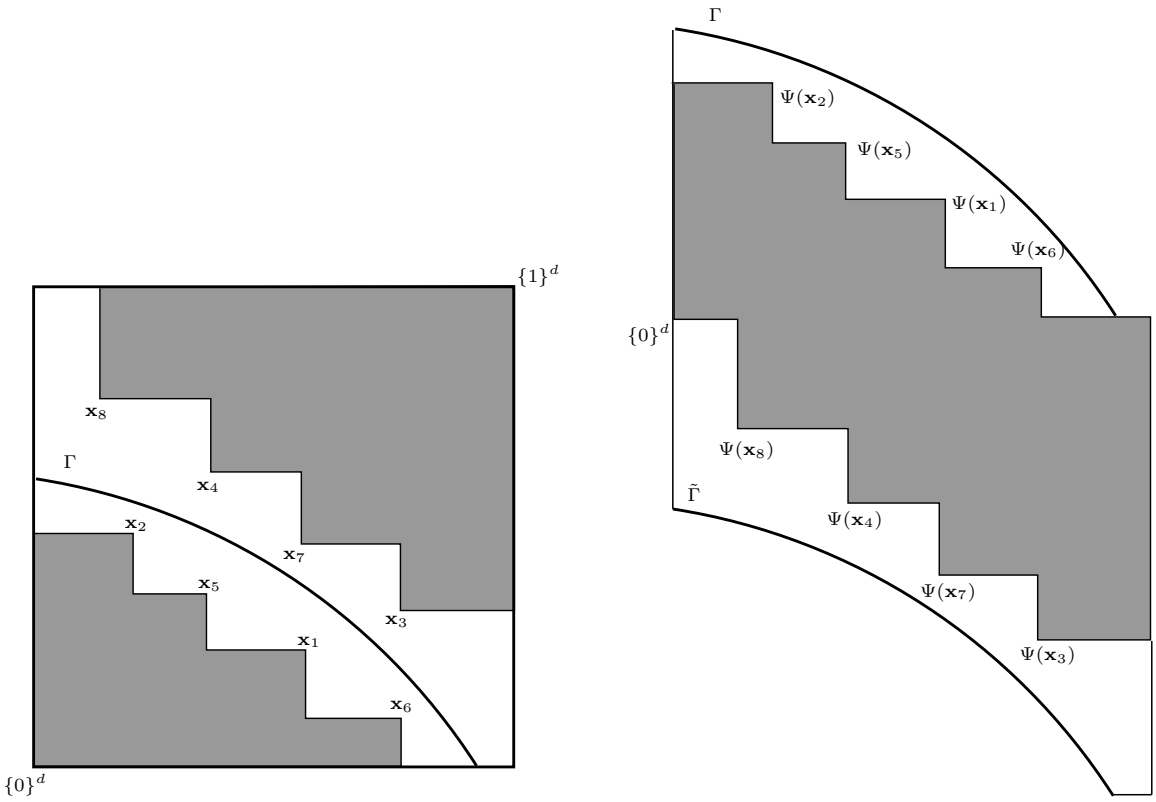


Figure 3.8: Illustration of the cut given in proof of Proposition 3.4. **Left**: Illustration of $[0,1]^d$ after some simulations. **Right**: Representation of $\tilde{\mathbb{U}}$ delimited by the set in gray, Γ and $\tilde{\Gamma}$. $\Psi(\mathbf{x}_i) = \mathbf{x}_i$ for $i \in \{3,4,7,8\}$.

Then, p_n^+ have the same distribution as $1 - p_n^-(1-p)$. For all $x \in [0,1]$, one has

$$\mathbb{P}(p_n^+ > x) = \mathbb{P}(p_n^-(1-p) \leq 1-x) = \begin{cases} 1 & \text{if } x < p \\ (1+p-x)^n & \text{if } p < x \leq 1 \\ 0 & \text{if } x > 1. \end{cases},$$

and

$$\mathbb{E}[p_n^+] = \mathbb{E}[1 - p_n^-(1-p)] = p + \frac{1}{n+1}[1-p^{n+1}].$$

It can be deduced that for all n,

$$\mathbb{E}[p_n^+ - p_n^-] = \frac{2}{n+1} - \frac{1}{n+1}\left(p^{n+1} + (1-p)^{n+1}\right).$$

□

Proof of Proposition 3.6. $\mathcal{F}_n$ will denote the natural filtration. One has $p_{n+1}^- + p_{n-1}^+ = (p_n^+ - p_n^-)\xi_{n+1} + X_{n+1} + p_n^-$. Since for all $k \geq 1, X_k$ is uniformly distributed on $[p_{k-1}^-, p_{k-1}^-]$ if $\mathcal{F}_n$ denotes the natural filtration associated to this sequence, it comes that

$$\begin{aligned} \mathbb{E}[p_{n+1}^- + p_{n+1}^+ | \mathcal{F}_n] &= (p_n^+ - p_n^-)\mathbb{E}[\xi_{n+1}|\mathcal{F}_n] + \mathbb{E}[X_{n+1}|\mathcal{F}_n] + p_n^-, \\ &= p + \mathbb{E}[X_{n+1}|\mathcal{F}_n] = p + (p_n^- + p_n^+)/2, \end{aligned}$$

the last equality holds since $\mathbb{E}[X_{n+1}|\mathcal{F}_n] = (p_n^- + p_n^+)/2$. By recursion, it can be deduced that

$$\begin{aligned}\mathbb{E}[X_{n+1}] &= \frac{1}{2}\mathbb{E}[p_n^- + p_n^+] = \frac{1}{2}(p + \mathbb{E}[X_n]) = \frac{p}{2} + \frac{p}{2^2} + \frac{\mathbb{E}[X_{n-1}]}{2^2}, \\ &= p\left(1 - \frac{1}{2^n}\right) + \frac{1}{2^{n+1}}.\end{aligned}$$

Since $X_{n+1} \in [0,1]$, it comes that $\mathrm{Var}[X_{n+1}] \leq \mathbb{E}[X_{n+1}^2] \leq \mathbb{E}[X_{n+1}] = p(1 - \frac{1}{2^n}) + \frac{1}{2^{n+1}}$. Besides

$$\begin{aligned}\mathbb{E}[p_{n+1}^+ - p_{n+1}^-|\mathcal{F}_n] &= (p_n^+ + p_n^-)\mathbb{E}[\xi_{n+1}|\mathcal{F}_n] - 2\mathbb{E}[X_{n+1}\xi_{n+1}|\mathcal{F}_n] + \mathbb{E}[X_{n+1}|\mathcal{F}_n] - p_n^-, \\ &= \frac{(p_n^+ + p_n^-)(p - p_n^-)}{p_n^+ - p_n^-} - \frac{p^2 - (p_n^-)^2}{p_n^+ - p_n^-} + \frac{p_n^+ + p_n^-}{2} - p_n^-, \\ &= \frac{p_n^+ - p_n^-}{2} + \frac{(p_n^+ - p)(p - p_n^-)}{p_n^+ - p_n^-}. \end{aligned} \tag{3.11}$$

Since $\mathbb{E}[p_{n+1}^+ - p_{n+1}^-] \geq \frac{1}{2}\mathbb{E}[p_n^+ - p_n^-]$, it comes by recursion that

$$\mathbb{E}[p_{n+1}^+ - p_{n+1}^-] \geq \frac{1}{2^{n+1}}.$$

Since $(p_n^+ - p)(p - p_n^-)$ is lower than $(p_n^+ - p_n^-)^2/4$, it can be deduced from (3.11) that

$$\mathbb{E}[p_{n+1}^+ - p_{n+1}^-|\mathcal{F}_n] \leq \frac{3}{4}(p_n^+ - p_n^-).$$

By recursion, it comes that $\mathbb{E}[p_n^+ - p_n^-] \leq \left(\frac{3}{4}\right)^n$. □

Proof of Corollary 3.2. Let $p = 0$, then $\mathbb{E}[p_{n+1}^+|\mathcal{F}_n] = \mathbb{E}[X_{n+1}|\mathcal{F}_n]$. Since $X_{n+1} \sim \mathcal{U}([0, p_n^+])$, it comes that

$$\mathbb{E}[p_{n+1}^+|\mathcal{F}_n] = p_n^+/2,$$

by recursion $\mathbb{E}[p_n^+] = 2^{-n}$. For $p = 1$, using $X_{n+1} \sim \mathcal{U}([p_n^-, 1])$ proves the result. □

The proof of Proposition 3.7 relies on the following Lemma that gives the value of the probability density function of a sum of independent beta distributed random variables.

Lemma 3.2 *Denote f_d the probability density function (pdf) of $\sum_{i=1}^d B_i$, supported over $[0,d]$, where the B_i are iid random variables following the $\mathcal{B}eta(1/q, 1)$ distribution. Then for all $x \in [0,1]$*

$$f_d(x) = C_{d,q}x^{d/q-1} \tag{3.12}$$

with $C_{1,q} = 1$ and for $d \geq 2$, $C_{d,q} = \frac{1}{q^d}\prod_{i=1}^{d-1} B(i/q, 1/q)$.

Proof of Lemma 3.2. We proceed by induction. For $d = 1$, the density given by (3.12) is the density of $\mathcal{B}eta(1/q, 1)$ distribution, that is

$$f_1(x) = \frac{1}{q}x^{1/q-1}\mathbb{1}_{\{0\leq x\leq 1\}}.$$

Assume there exists $k \in \mathbb{N}^*$ such that for all $1 \leq j \leq k$, f_j is the density of $\sum_{i=1}^{j} B_i$ (only for $x \in [0,1]$). Denote $f \star g$ the convolution of functions f and g. Then, for $x \in [0,1]$,

$$\begin{aligned} f_{k+1}(x) = (f_k \star f_1)(x) &= \int_0^x f_k(t) f_1(x-t) dt, \\ &= \int_0^x \frac{1}{q^k} \left[\prod_{i=1}^{k-1} B(i/q, 1/q) \right] t^{k/q-1} \frac{1}{q} (x-t)^{1/q-1} dt, \\ &= \frac{1}{q^{k+1}} \prod_{i=1}^{k-1} B(i/q, 1/q) \int_0^x x^{1/q-1} t^{k/q-1} (1-t/x)^{1/q-1} dt, \\ &= C_{k+1,q} x^{(k+1)/q-1} \end{aligned}$$

which proves the validity of (3.12). □

Proof of Proposition 3.7. Set $X_i = (X_i^1, \ldots, X_i^j)$. Since $\mathbb{U}_n^- \subset \mathbb{U}^-$, one has

$$\begin{aligned} d_{H,q}(\mathbb{U}_n^-, \mathbb{U}^-) &= \max\Big(\sup_{\mathbf{y} \in \mathbb{U}_n^-} \inf_{\mathbf{x} \in \mathbb{U}^-} \|\mathbf{x}-\mathbf{y}\|_q ; \sup_{\mathbf{x} \in \mathbb{U}^-} \inf_{\mathbf{y} \in \mathbb{U}_n^-} \|\mathbf{x}-\mathbf{y}\|_q \Big), \\ &= \sup_{\mathbf{x} \in \mathbb{U}^-} \inf_{\mathbf{y} \in \mathbb{U}_n^-} \|\mathbf{x}-\mathbf{y}\|_q = \inf_{\mathbf{y} \in \mathbb{U}_n^-} \|\{1\}^d - \mathbf{y}\|_q, \\ &= \inf_{\mathbf{y} \in \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}} \|\{1\}^d - \mathbf{y}\|_q. \end{aligned}$$

(1). Assume $0 < q < \infty$. For $t \in [0, d^{1/q}]$, using the fact that if X is uniformly distributed on $[0,1]$ so is $1-X$ we have

$$\mathbb{P}(d_{H,q}(\mathbb{U}_n^-, \mathbb{U}^-) \leq t) = 1 - \left(1 - \mathbb{P}\left(\sum_{j=1}^{d} (X_1^j)^q \leq t^q \right) \right)^n.$$

Let $(\alpha_n)_{n \geq 1}$ be a sequence of real numbers such that $\alpha_n \to +\infty$ as $n \to +\infty$. Hence,

$$\mathbb{P}(\alpha_n d_{H,q}(\mathbb{U}_n^-, \mathbb{U}^-) \leq t) = 1 - \left(1 - \mathbb{P}\left(\sum_{i=1}^{d} B_i \leq \frac{t^q}{\alpha_n^q} \right) \right)^n \tag{3.13}$$

where each $B_i = X_i^q \overset{\mathcal{L}}{\sim} \mathcal{B}eta(1/q, 1)$. From Lemma 3.2, for $u \in [0,1]$,

$$\mathbb{P}\left(\sum_{i=1}^{d} B_i \leq u \right) = \int_0^u C_{d,q} x^{d/q-1} dx = \frac{q}{d} C_{d,q} u^{d/q},$$

where $C_{d,q} = \frac{1}{q^d} \prod_{i=1}^{d-1} B(i/q, 1/q)$. Therefore, for n large enough

$$\mathbb{P}(\alpha_n d_{H,q}(\mathbb{U}_n^-, \mathbb{U}^-) \leq t) = 1 - \left(1 - \frac{q}{d} C_{d,q} \frac{t^d}{\alpha_n^d} \right)^n.$$

Denoting $A_{1,q} = 1$ and for $d \geq 2$, $A_{d,q} = \frac{1}{dq^{d-1}} \prod_{i=1}^{d-1} B(i/q, 1/q)$, and choosing $\alpha_n = (n A_{d,q})^{1/d}$ it comes

$$\mathbb{P}\left((A_{d,q} n)^{1/d} d_{H,q}(\mathbb{U}_n^-, \mathbb{U}^-) \leq t \right) \underset{n \to +\infty}{\longrightarrow} 1 - e^{-t^d}.$$

(2). Assume now $q = \infty$. Let $(\beta_n)_{n\geq 1}$ be a sequence of real numbers such that $\beta_n \to +\infty$ as $n \to +\infty$. Then for all $t \in]0,1[$

$$\begin{aligned}\mathbb{P}(\beta_n d_H^{\infty}(\mathbb{U}_n^-, \mathbb{U}^-) > t) &= (1 - \mathbb{P}(\|\mathbf{X}_1\|_{\infty} \leq t/\beta_n))^n = \Big(1 - \mathbb{P}(\max_{j=1,\dots,d} X_1^j \leq t/\beta_n)\Big)^n, \\ &= \Big(1 - t^d/\beta_n^d\Big)^n,\end{aligned}$$

choosing $\beta_n = n^{1/d}$, it comes $\mathbb{P}(n^{1/d} d_H^{\infty}(\mathbb{U}_n^-, \mathbb{U}^-) > t) \underset{n\to+\infty}{\longrightarrow} e^{-t^d}$, and

$$n^{1/d} d_H^{\infty}(\mathbb{U}_n^-, \mathbb{U}^-) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{W}(1,d).$$

□

Proof of Proposition 3.8. Let $(\mathbf{X}_n)_{n\geq 1}$ be a sequence of iid random variables uniformly distributed on $[0,1]^d$. Let $U = (U^1, \dots, U^d)$ be uniformly distributed on $[0,1]^d$ and independent of the sample $(\mathbf{X}_n)_{n\geq 1}$, then

$$\begin{aligned}\mathbb{E}[\mu([0,1]^d \backslash \mathbb{U}_n^-)] &= \mathbb{E}[\mathbb{E}[\mathbb{1}_{\mathbf{U}\in\mathbb{U}_n}|\mathbf{U}]], \\ &= \mathbb{E}[\mathbb{E}[\mathbb{1}_{\mathbf{U}\not\preceq\mathbf{X}_1,\dots,\mathbf{U}\not\preceq\mathbf{X}_n}|\mathbf{U}]], = \mathbb{E}[\mathbb{E}[\mathbb{1}_{\mathbf{U}\not\preceq\mathbf{X}_1}|\mathbf{U}] \cdots \mathbb{E}[\mathbb{1}_{\mathbf{U}\not\preceq\mathbf{X}_n}|\mathbf{U}]], \\ &= \mathbb{E}[\mathbb{E}[\mathbb{1}_{\mathbf{U}\not\preceq\mathbf{X}_1}|\mathbf{U}]^n] = \mathbb{E}[\Big(1 - (1-U^1)\cdots(1-U^d)\Big)^n], \\ &= \mathbb{E}[\Big(1 - U^1 \cdots U^d\Big)^n]. \end{aligned} \tag{3.14}$$

Using the fact that $-\log(U^i) \overset{\mathcal{L}}{\sim} \mathcal{E}xp(1)$, it is easy to see that $\prod_{i=1}^d U^i \overset{\mathcal{L}}{=} \exp(-G_d)$, where $G_d \sim \mathcal{G}amma(d,1)$ with density function

$$f(x) = \frac{x^{d-1}e^{-x}}{(d-1)!}.$$

We then easily get that the density function of $\prod_{i=1}^d U^i$ is given by

$$f_d(x) = \frac{(-\log(x))^{d-1}}{(d-1)!}\mathbb{1}_{\{x\in[0,1]\}}. \tag{3.15}$$

From (3.14) and (3.15), it comes

$$\mathbb{E}[\mu([0,1]^d \backslash \mathbb{U}_n)] = \mathbb{E}[\Big(1 - U^1 \cdots U^d\Big)^n] = \int_0^1 (1-u)^n \frac{(-\log(u))^{d-1}}{(d-1)!} du.$$

One has

$$\frac{n(d-1)!}{\log(n)^{d-1}}\mathbb{E}[\mu([0,1]^d \backslash \mathbb{U}_n)] = \frac{n}{\log(n)^{d-1}} \int_0^1 (1-u)^n (-\log(u))^{d-1} du,$$

and the substitution $u = x/n$ gives

$$\begin{aligned}\frac{n}{\log(n)^{d-1}} \int_0^1 (1-u)^n (-\log(u))^{d-1} du &= \frac{1}{\log(n)^{d-1}} \int_0^{+\infty} (1-x/n)^n (-\log(x/n))^{d-1} \mathbb{1}_{\{x\leq n\}} dx, \\ &= \int_0^{+\infty} (1-x/n)^n (1 - \log(x)/\log(n))^{d-1} \mathbb{1}_{\{x\leq n\}} dx, \\ &\underset{n\to+\infty}{\longrightarrow} 1.\end{aligned}$$

The last equation holds by the dominated convergence theorem apply to $f_n(x) = (1-x/n)^n (1 - \log(x)/\log(n))^{d-1} \mathbb{1}_{\{0\leq x\leq n\}} \leq \exp(-x)\mathbb{1}_{\{x\geq 0\}}$.

Proof of Proposition 3.9. At step $k \geq 1$ denote $\tilde{p}_k^- = \mu(\mathbb{U}^-(\mathbf{X}_1)) + \sum_{j=2}^{k} \sum_{l=1}^{2^j} \mu(\mathbb{U}^-(\mathbf{X}_{j,l}) \cap E_{j,l})$. Conditionning to $\mathcal{F}_{k-1} = \sigma(\mathbf{X}_{j,l}, 1 \leq j \leq k-1, 1 \leq l \leq 2^j)$ it comes

$$\begin{aligned}
\mathbb{E}[\tilde{p}_k^- | \mathcal{F}_{k-1}] &= \mu(\mathbb{U}^-(\mathbf{X}_1)) + \sum_{j=2}^{k-1} \sum_{l=1}^{2^j} \mu(\mathbb{U}^-(\mathbf{X}_{k,l}) \cap E_{k,l}) + \sum_{l=1}^{2^k} \mathbb{E}[\mu(\mathbb{U}^-(\mathbf{X}_{k,j}) \cap E_{k,j}) | \mathcal{F}_{k-1}], \\
&= p_{k-1}^- + \sum_{l=1}^{2^k} \mathbb{E}[\mu(\mathbb{U}^-(\mathbf{X}_{k,l}) \cap E_{k,l}) | \mathcal{F}_{k-1}], \\
&= p_{k-1}^- + \sum_{j=1}^{2^n} \frac{1}{4} \mathbb{E}[\mu(E_{k,l}) | \mathcal{F}_{k-1}], \\
&= p_{k-1}^- + \frac{1}{4}(1 - p_{k-1}^-),
\end{aligned}$$

and then $\mathbb{E}[\tilde{p}_k^-] = \frac{1}{4} + \frac{3}{4}\mathbb{E}[\tilde{p}_{k-1}^-]$. By induction, it comes

$$\begin{aligned}
\mathbb{E}[\tilde{p}_k^-] &= \frac{1}{4} \sum_{j=0}^{k-1} + (3/4)^j, \\
&= 1 - (3/4)^k.
\end{aligned}$$

Since $n = 2^k - 1$, it comes

$$\begin{aligned}
\mathbb{E}[p_n^-] = \mathbb{E}[\tilde{p}_{2^k-1}^-] &= 1 - (3/4)^{\frac{\log(n+1)}{\log 2}}, \\
&= 1 - \frac{1}{(n+1)^{-\log(3/4)/\log 2}}.
\end{aligned}$$

□

Proof of Proposition 3.10. Following the proof of Proposition 3.9, the results is straightforward. □

Proof of Proposition 3.11. Let $\mathbf{u} \in \Gamma$ and $\mathbf{v} \in [0,1]^d$ such that $\mathbf{u} \prec \mathbf{v}$. Assume $\mathbf{v} \in \Gamma$. There exist $\varepsilon > 0$ such that $\mathbf{B}((\mathbf{u}+\mathbf{v})/2, \varepsilon) \subset \Gamma$. That implies $\mu(\Gamma) > 0$, which is impossible by Assumption 3.2. Then $\mathbf{v} \notin \Gamma$. □

Proof of Theorem 3.1. **(1)**. Obvious since $h_{\mathbf{X}}(\mathbf{u}) \leq h_{\mathbf{X}}(\mathbf{v})$ for all $\mathbf{X} \in \Xi_n^+$ and for all $\mathbf{u}, \mathbf{v} \in [0,1]^d$ such that $\mathbf{u} \preceq \mathbf{v}$.
(2), (3). By construction $\mathbb{U}_n^- \subset F_n \subset [0,1]^d \backslash \mathbb{U}_n^+$.
(4). Since $\mathbb{U}^-$ is regular (by convexity) and $\mathbb{U}^+$ is regular (by Lemma 3.3 below), the result is a consequence of Equation (3.4).

Lemma 3.3 *Let K be a compact, convex subset of $[0,1]^d$ such that for all $\mathbf{y} \in K$ there is no $\mathbf{x} \in K^c = [0,1]^d \backslash K$ such that $\mathbf{x} \preceq \mathbf{y}$. Then K^c is (α, γ)-regular.*

Proof of Lemma 3.3. Since K is a compact convex set, there exists $\alpha, \gamma > 0$ such that K is (α, γ)-regular [4]. Given $\mathbf{x} \in K^c$, since K is closed convex the projection of $\mathbf{x}$ on K, denoted $P_K(\mathbf{x})$, is unique. Let $0 \leq \varepsilon \leq \gamma$ and $\mathbf{x} \in K^c$, then

$$\begin{aligned}
\mu(\mathbf{B}(\mathbf{x}, \varepsilon) \cap K^c) &\geq \mu(\mathbf{B}(P_K(\mathbf{x}), \varepsilon) \cap K^c), \\
&\geq \mu(\mathbf{B}(P_K(\mathbf{x}), \varepsilon) \cap K), \text{ since } K \text{ is convex} \\
&\geq \alpha\mu(\mathbf{B}(P_K(\mathbf{x}), \varepsilon)), \text{ since } K \text{ is } (\alpha, \gamma)\text{-regular} \\
&\geq \alpha\mu(\mathbf{B}(\mathbf{x}, \varepsilon)),
\end{aligned}$$

then K^c is (α, γ)-regular. □

Proof of Corollary 3.3. Since g is differentiable at p, the Delta method and Proposition 3.5 imply that

$$\begin{aligned}
n(g(p) - g(p_n^-)) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)), \\
n(g(p_n^+) - g(q)) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)).
\end{aligned}$$

Since q_p is the p-order quantile of $g(\mathbf{X})$, it comes that $\mathbb{P}(g(X) \leq q_p) = p$. Since g is continuous and strictly increasing, then

$$\mathbb{P}(g(X) \leq q_p) = \mathbb{P}(X \leq g^{-1}(q_p)).$$

Moreover $X \sim \mathcal{U}([0,1])$, then $\mathbb{P}(X \leq g^{-1}(q_p)) = g^{-1}(q_p) = p$. It can be deduced that $q_p = g(p)$. Then, the two last equations becomes

$$\begin{aligned}
n(q_p - g(p_n^-)) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)), \\
n(g(p_n^+) - q_p) &\xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{E}xp(1/g'(p)).
\end{aligned}$$

□

3.8 Appendix

3.8.1 Computing hypervolumes (deterministic bounds)

The computation of bounds (p_n^-, p_n^+) defining in (3.3) can be conducted exactly or using simulation, in function of the dimension.

An exact method in dimension $d = 2$

Consider a design $\mathbf{x}_1, \ldots, \mathbf{x}_n \in [0,1]^2$. The first stage is to order the element according to their first (or second) coordinate, such that $x_1^1 < \cdots < x_n^1$. Since no design element is dominated by another one, then $x_i^2 > x_{i+1}^2$ for $i \in \{1, \ldots, n-1\}$. The point $\mathbf{x}_1$ delimit a first rectangle P_1 with the following vertices:

$$\begin{pmatrix} 0 & 0 \\ x_1^1 & 0 \\ 0 & x_1^2 \\ x_1^1 & x_1^2 \end{pmatrix},$$

such that $\mu(P_1) = x_1^1 x_1^2$. For all $i \in \{2, \ldots, n\}$ a new rectangle P_i can be defined with the following vertices:

$$\begin{pmatrix} x_{i-1}^1 & 0 \\ x_{i-1}^1 & x_i^2 \\ x_i^1 & 0 \\ x_i^1 & x_i^2 \end{pmatrix},$$

such that $\mu(P_i) = (x_i^1 - x_{i-1}^1)x_i^2$. The second stage consist to compute the volume of each rectangles:

$$p_n^- = x_1^1 x_1^2 + \sum_{i=2}^{n} (x_i^1 - x_{i-1}^1) x_i^2.$$

To compute p_n^+ it is enough to transform each $\mathbf{x}_i$ into $1 - \mathbf{x}_i$, to compute $\tilde{p}_n^+$ with the previous approach. Then $p_n^+ = 1 - \tilde{p}_n^+$.

An accelerated Monte Carlo method in upper dimensions

A sweepline algorithm described in [16] allows to compute the bounds in any dimension, but at an exponential cost. An alternative approach is using a Monte Carlo method. Considering an iid uniform sample $\bar{\mathbf{U}}_N = \mathbf{U}_1, \ldots, \mathbf{U}_N$ over $[0,1]^d$, (p_n^-, p_n^+) can be estimated by

$$\hat{p}_n^- = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\mathbf{U}_i \in \mathbb{U}_n^-\}},$$

$$\hat{p}_n^+ = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{\mathbf{U}_i \in \mathbb{U}_n^+\}}.$$

The computation can be strongly accelerated by adapting the order of evaluation to the monotonic context, using the following algorithm (easily adaptable to estimating p_n^+).

Algorithm 3.1 Estimation of p_n^- by accelerated Monte Carlo

$\tilde{p}_n^- \leftarrow 0$, $\mathbf{V} = \bar{\mathbf{U}}_n$, $\bar{\mathbf{U}}_N^- = \bar{\mathbf{U}}_n \cap \mathbb{U}_n^-$
for $i : 1$ to $Card(\bar{\mathbf{U}}_N^-)$ **do**
  $\mathbf{u} \in \arg\max_{\mathbf{x} \in \bar{\mathbf{U}}_N^-} \prod_{j=1}^{d} x_j$
  $\Xi_n^- \leftarrow \Xi_n^- \backslash \mathbf{u}$
  $\tilde{p}_n^- \leftarrow \tilde{p}_n^- + Card(\{\mathbf{U} \in \mathbf{V} : \ \mathbf{U} \preceq \mathbf{u}\})$
  $\mathbf{V} \leftarrow \mathbf{V} \backslash \{\mathbf{U} \in \mathbf{V} : \ \mathbf{U} \preceq \mathbf{u}\}$
end for
return $\tilde{p}_n^- / N$

3.8.2 Adapted Support Vector Machines

Usual situations: a reminder

Linear situation. Consider a situation where some available data $D_n = (\mathbf{X}_i, y_i)_{1 \leq i \leq n}$ where $\mathbf{X}_i \in [0,1]^d$ and $y_i = sign(h(\mathbf{X}_i))$, with $sign(x) = 1$ if $x > 0$ and -1 otherwise, can be perfectly

separated according to y by an hyperplane h defined by

$$h(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \mathbf{x}, \tag{3.16}$$

with $(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^d$. In this linear framework, an infinity of hyperplanes can separate perfectly the data. Therefore it is needed to add some constraints to make a unique (optimal) choice of hyperplane. Define the distance of $\mathbf{x}$ from Γ :

$$\Delta(\mathbf{x}, \Gamma) = \inf_{\mathbf{y} \in \Gamma} \|\mathbf{x} - \mathbf{y}\| = |\beta_0 + \boldsymbol{\beta}^T \mathbf{x}| / \|\boldsymbol{\beta}\|,$$

from (3.16). The chosen hyperplane is the solution of the following problem:

$$\begin{cases} \max\limits_{\beta_0, \boldsymbol{\beta}} & m_{\beta_0, \boldsymbol{\beta}} \\ \min\limits_{i=1,\dots,n} & \dfrac{|\beta_0 + \boldsymbol{\beta}^T \mathbf{X}_i|}{\|\boldsymbol{\beta}\|} \geq m. \end{cases} \tag{3.17}$$

After the substitution $\mathbf{w} = \frac{\boldsymbol{\beta}}{m\|\boldsymbol{\beta}\|}$ and $w_0 = \frac{\beta_0}{m\|\boldsymbol{\beta}\|}$ [65], the problem (3.17) becomes:

$$\begin{cases} \min\limits_{w_0, \mathbf{w}} & \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{for } i = 1, \dots, n & y_i(w_0 + \mathbf{w}^T \mathbf{X}_i) \geq 1. \end{cases} \tag{3.18}$$

A number $d+1$ of parameters associated to n linear constraints has to be estimated. Since the problem (3.18) is quadratic with linear constraints, a unique solution can be found. The dual form of the problem is given by the following optimisation problem

$$\begin{cases} \max\limits_{\mathbf{a}} & \displaystyle\sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i,j=1}^{n} a_i a_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \displaystyle\sum_{i=1}^{n} a_i y_i = 0 \\ \mathbf{a} \succeq 0. \end{cases} \tag{3.19}$$

where $\mathbf{a} = (a_1, \dots, a_n)$ is a Lagrange multiplier. Let $\mathbf{a} = (a_1, \dots, a_n)$ be the solution of problem (3.19). The computer model g can be estimated by

$$\widehat{g}_n(\mathbf{x}) = w_0 + \sum_{i=1}^{n} a_i y_i \mathbf{X}_i^T \mathbf{x}. \tag{3.20}$$

Table 3.3 suggest to solve the primal problem if $d \leq n$, and the dual problem in the other case.

	primal	dual
Number of parameters	$d+1$	n
Number of constraints	n	$n+1$

Table 3.3: Numbers of constraints and dimension of parameters to be estimated for the primal and dual problems in the usual SVM framework.

Nonlinear situation. Assume now that D_n cannot be separated by a hyperplane but by a nonlinear surface. Let K be a symmetrical positive definite kernel. From Mercer's theorem there exists a transformation $h : [0,1]^d \to \mathcal{H}$ where $\mathcal{H}$ is an Hilbert space with the inner product

$$\langle h(\mathbf{x}), h(\mathbf{y})\rangle_{\mathcal{H}} = K(\mathbf{x}, \mathbf{y}). \tag{3.21}$$

such that the data can be linearly separated. Then g can be written as

$$g(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T h(\mathbf{x}),$$

and using (3.20) and (3.21), it can be estimated by:

$$\widehat{g}_n(\mathbf{x}) = w_0 + \sum_{i=1}^{n} a_i y_i K(\mathbf{X}_i, \mathbf{x}).$$

The optimization problem (3.19) becomes

$$\begin{cases} \max\limits_{\mathbf{a}} & \sum\limits_{i=1}^{n} a_i - \dfrac{1}{2}\sum\limits_{i,j=1}^{n} a_i a_j y_i y_j K(\mathbf{X}_i, \mathbf{x}_j) \\ \sum\limits_{i=1}^{n} a_i y_i = 0 \\ \mathbf{a} \succeq 0 \end{cases}$$

or

$$\begin{cases} \min\limits_{\mathbf{a}} & \frac{1}{2}\mathbf{a}^T \widehat{K}\mathbf{a} - \mathbf{d}^T\mathbf{a} \\ A\mathbf{a} \overset{*}{\geq} \mathbf{c}, \end{cases}$$

where the symbol $\overset{*}{\geq}$ means that the first constraint is an equality constraint and the other are inequalities constraints, and

$$A = \begin{pmatrix} y_1 \cdots y_n \\ I_n \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbf{R}^{n+1}, \quad \mathbf{d} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbf{R}^n$$

and $\widehat{K}_{i,j=1,\ldots,n} = y_i y_j K(\mathbf{X}_i, \mathbf{X}_j)$, with I_n the identity matrix.

Monotonic SVM

Let g be defined by (3.16) and assume now that g is globally increasing. Let $\mathbf{x} = (x_1, \ldots, x_d) \in [0,1]^d$ and $\mathbf{x}^i = (x_1, \ldots, x_{i-1}, x_i + \eta, x_{i+1}, \ldots, x_d)$ with $\eta > 0$, then $\mathbf{x} \preceq \mathbf{x}^i$ and $g(\mathbf{x}) \leq g(\mathbf{x}^i)$. It comes that

$$g\left(\mathbf{x}^i\right) - g(\mathbf{x}) \geq 0 \Rightarrow \beta_i \geq 0,$$

for all $i = 1, \ldots, d$. The problem (3.18) becomes:

$$\begin{cases} \min\limits_{w_0, \mathbf{w}} & \dfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{for } i = 1, \ldots, n & y_i(w_0 + \mathbf{w}^T\mathbf{X}_i) \geq 1 \cdot \\ \text{for } i = 1, \ldots, d & w_i \geq 0 \end{cases} \tag{3.22}$$

The associated Lagrangian $\mathcal{M}(\mathbf{w}, w_0, \mathbf{a}, \mathbf{b})$ can be written as

$$\mathcal{M}(\mathbf{w}, w_0, \mathbf{a}, \mathbf{b}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} a_i[y_i(w_0 + \mathbf{w}^T\mathbf{X}_i) - 1] - \mathbf{b}^T\mathbf{w},$$

where $\mathbf{a}$ and $\mathbf{b}$ are the Lagrangian multipliers. The dual problem becomes

$$\begin{cases} \max\limits_{\mathbf{a},\mathbf{b}} \sum\limits_{i=1}^{n} a_i - \frac{1}{2}\sum\limits_{i,j=1}^{n} a_i a_j y_i y_j \mathbf{X}_i^T\mathbf{X}_j - \mathbf{b}^T\sum\limits_{i=1}^{n} a_i y_i \mathbf{X}_i - \frac{1}{2}\|\mathbf{b}\|^2 \\ \sum\limits_{i=1}^{n} a_i y_i = 0 \\ \mathbf{a} \succeq 0 \\ \mathbf{b} \succeq 0. \end{cases}.$$

Table 3.4 shows there is always less constraints and parameters to estimate in the primal problem than in the dual problem. In practice, it is more interesting to solve the primal problem.

	primal	dual
Number of parameters	$d+1$	$d+n$
Number of constraints	$d+n$	$d+n+1$

Table 3.4: Numbers of constraints and dimension of parameters to be estimated for the primal and dual problems in the monotonic SVM framework.

The problem (3.22) can then be rewritten.

$$\begin{cases} \min\limits_{w_0,\mathbf{w}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ & AW \geq \mathbf{c} \end{cases}$$

where

$$A = \begin{pmatrix} A_{1,1} \\ A_{2,1} \end{pmatrix}, \quad W = \begin{pmatrix} \mathbf{w} \\ w_0 \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} \mathbf{1}_n \\ \mathbf{0}_d \end{pmatrix} \in \mathbf{R}^{n+d},$$

with

$$A_{1,1} = \begin{pmatrix} y_1\mathbf{X}_1^T & y_1 \\ \vdots & \vdots \\ y_n\mathbf{X}_n^T & y_n \end{pmatrix} \in \mathcal{M}_{n\times(d+1)}(\mathbb{R}), \quad A_{2,1} = \begin{pmatrix} I_d & \mathbf{0}_d \end{pmatrix} \in \mathcal{M}_{d\times(d+1)}(\mathbb{R}),$$

and I_d the identity matrix in $\mathbb{R}^d$, $\mathbf{1}_n^T = (1, \ldots, 1) \in \mathbb{R}^n$ and $\mathbf{0}_d^T = (0, \ldots, 0) \in \mathbb{R}^d$.

Chapter 4

Sequential adaptive estimation of limit state probability in a monotonic Monte Carlo framework

Résumé Dans ce chapitre, on considère le problème d'estimation de la probabilité de dépassement de seuil définie par $p = P(g(\mathbf{X}) \leq 0)$ où g est une fonction monotone, coûteuse en temps de calcul et de type boîte noire et $\mathbf{X}$ un vecteur aléatoire d dimensionnel. Ce cadre est typique des problèmes rencontrés en fiabilité structurale. Tirant parti de la propriété de monotonie, des bornes déterministes pour p peuvent être calculées à partir d'un échantillon. Un estimateur statistique de p a été étudié dans [16], qui présente une forte réduction de variance par rapport à une approche Monte Carlo direct. En revanche, cet estimateur est biaisé et est basé sur un tirage séquentiel naïf. Dans ce chapitre, une nouvelle famille d'estimateurs de p est construite à partir d'un tirage d'importance adaptatif.

Abstract In this chapter, the estimation of a limit state probability defined by $p = P(g(\mathbf{X}) \leq 0)$ is considered, where g is a monotonic, time-consuming black-box function and $\mathbf{X}$ is a d-dimensional random vector. This framework occurs typically in structural reliability problems. Based on the monotonicity property, deterministic bounds around p can be computed from designs of numerical experiments. A statistical estimator of p was investigated in [16]. It leads to high variance reduction compared to an usual Monte Carlo approach. However, it suffered from bias and was based on naive nested uniform sampling. In this chapter, a new family of sequential estimators of p is built from an adaptive importance sampling scheme.

4.1 Introduction

A situation frequently occurring in structural reliability studies is the estimation of a so-called limit state probability [77], or failure probability, defined by

$$p = P(g(\mathbf{X}) \leq 0), \tag{4.1}$$

where g is a deterministic function and $\mathbf{X}$ is an input random vector of dimension d. Most often, g is a time-consuming, black-box computer model. The parameter p should be estimated in a non-intrusive way, by means of a well-tailored design. In this context, Monte Carlo variance reduction techniques are currently used to decrease the size of the design and limit the computational burden. See [86] for a recent survey of such techniques.

The use of form constraints on g may improve the convergence rate of the estimate. When g is assumed to be monotonic according to Definition 2.1, it can be shown that the limit state surface $\Gamma := \{\mathbf{x} \in \mathbb{R}^d, g(\mathbf{x}) = 0\}$ can be contoured by two simply connected sets [38]. As discussed in Chapter 2, two bounds for p can be obtained from a sequential sampling scheme. Properties of these bounds have been obtained in Chapter 3. This present chapter aims at improving this approach by adopting an adaptive importance strategy in the sequential sampling of each $\mathbf{X}_i$. A family of unbiased estimators of p is provided and studied.

This chapter is organised as follows. In Section 4.2, it is recalled the framework and the main results previously obtained as well as a general estimate of p. A theoretical discussion on this general estimate is conducted in Section 4.3. An optimal choice of each importance distributions, is conducted. Numerical experiments are discussed in Section 4.4. The technical proofs are postponed to the end of the chapter.

4.2 Framework and previous results

Recall the following assumptions.

Assumption 4.1 *The function g is globally increasing, that is for all $\mathbf{u}, \mathbf{v} \in [0,1]^d$ such that $\mathbf{u} \preceq \mathbf{v}$, then $g(\mathbf{u}) \leq g(\mathbf{v})$.*

Assumption 4.2 *The input random variable $\mathbf{X}$ is uniformly distributed on $[0,1]^d$.*

Assumption 4.3 *The limit surface $\Gamma = \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = 0\}$ is simply connex and $\mu(\Gamma) = 0$ with μ the Lebesgue measure on $\mathbb{R}^d$.*

Further, recall the notations given in Chapter 3.

$$\begin{aligned}\mathbb{U}^- &:= \{\mathbf{x} \in [0,1]^d : \ g(\mathbf{x}) \leq 0\},\\ \mathbb{U}^+ &:= \{\mathbf{x} \in [0,1]^d : \ g(\mathbf{x}) > 0\},\end{aligned}$$

and $p = \mathbb{P}(\mathbf{X} \in \mathbb{U}^-) \in]0,1[$. Let $A \subset [0,1]^d$, define

$$\mathbb{U}^-(A) := \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^-} \{\mathbf{u} \in [0,1]^d : \ \mathbf{u} \preceq \mathbf{x}\}, \tag{4.2}$$

$$\mathbb{U}^+(A) := \bigcup_{\mathbf{x} \in A \cap \mathbb{U}^+} \{\mathbf{u} \in [0,1]^d : \ \mathbf{u} \succeq \mathbf{x}\}. \tag{4.3}$$

Obviously

$$\begin{aligned}&\mathbb{U}^-(A) \subset \mathbb{U}^- \ \subset \ [0,1]^d \backslash \mathbb{U}^+(A) \\ \text{and} \quad &\mu(\mathbb{U}^-(A)) \leq p \ \leq \ 1 - \mu(\mathbb{U}^+(A)).\end{aligned} \tag{4.4}$$

Definition 4.1 *Let $S \subset [0,1]^d$. The set S is monotonic if for all $\mathbf{u}, \mathbf{v} \in S$, $\mathbf{u}$ is not strictly dominated by $\mathbf{v}$.*

Remark 4.1 *Any limit state surface defined by $\{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\}$, $(q \in \mathbb{R})$ is a monotonic surface. This feature will appear as an important constraint further in the text.*

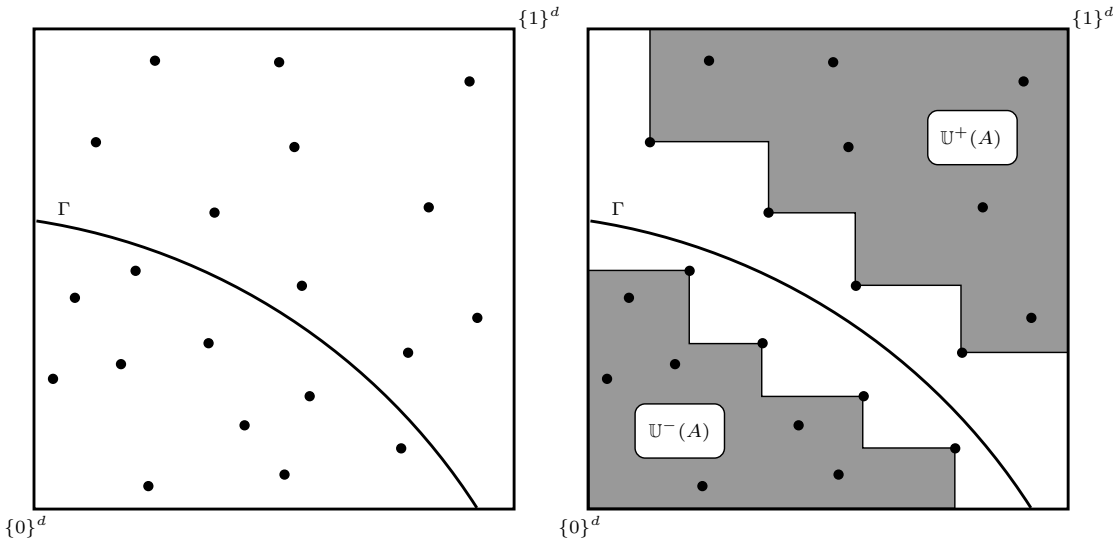


Figure 4.1: Illustration for $d = 2$. **Left**: let A be a set of points in $[0,1]^d$ represented by the black dots. **Right**: Below (resp. above) the limit state Γ, the set in gray represents $\mathbb{U}^-(A)$ (resp. $\mathbb{U}^+(A)$).

To estimate accurately the probability of $\{g(\mathbf{X}) \leq 0\}$, a sequential strategy are well-tailored when the total number of evaluations by g is limited by a predetermined computational budget. The pioneer [16] studies the simple case of a sequence of random vectors $(\mathbf{X}_k)_{k\geq 1}$ uniformly distributed on the non-dominated set $\mathbb{U}_{k-1} = [0,1]^d \backslash \left(\mathbb{U}^-_{k-1} \cup \mathbb{U}^+_{k-1}\right)$, where

$$\begin{aligned}\mathbb{U}^-_{k-1} &= \mathbb{U}^-(\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}),\\ \mathbb{U}^+_{k-1} &= \mathbb{U}^+(\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}).\end{aligned}$$

Although this first approach may have interest in a rough exploring step of $[0,1]^d$. It leads to to a consistent estimator $\check{p}_n$ of p. The variance of $\check{p}_n$ is significantly smaller than the variance of the usual Monte Carlo estimator. This variance is given by

$$Var[\check{p}_n] = 1/\sum_{k=1}^{n} \mathbb{E}\left[\tilde{\omega}_k(p)\right], \tag{4.5}$$

where $\tilde{\omega}_k(p) = [(p - p^-_{k-1})(p^+_{k-1} - p)]^{-1}$ and

$$\begin{aligned}p^-_{k-1} &= \mu(\mathbb{U}^-_{k-1}),\\ p^+_{k-1} &= 1 - \mu(\mathbb{U}^+_{k-1}).\end{aligned}$$

From (4.4), it comes $p^-_{k-1} \leq p \leq p^+_{k-1}$. In [16], it is proved that $\check{p}_n$ is a biased asymptotically normal estimator of p. In addition, the bias increases with d and a bootstrap procedure is studied in order to decrease the bias.

Denote $\mathcal{F}_n = \sigma(\mathbf{X}_1, \ldots, \mathbf{X}_n)$. As discussed in Section 2.3.2, an initialisation step provides $\mathbb{U}_0 \subsetneq [0,1]^d$ and p^-_0, p^+_0 in $]0,1[$.

Following [16], a general unbiased estimator of p can be easily constructed. Denote by $(\widetilde{\omega}_k)_{k\geq 1}$ a sequence of positive weights and for all $n \geq 1$ and for all $1 \leq k \leq n$ denote

$$\omega_{k,n} = \frac{\widetilde{\omega}_k}{\sum_{j=1}^{n} \widetilde{\omega}_j}.$$

Let f_{k-1} be the probability density function of $\mathbf{X}_k$. Set

$$\widetilde{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^{-} + \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^-\}} \right). \tag{4.6}$$

It comes

$$\begin{aligned} \mathbb{E}[\widetilde{p}_n] &= \sum_{k=1}^{n} \omega_{k,n} \mathbb{E}[p_{k-1}^{-} + \mathbb{E}[\frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^-\}} | \mathcal{F}_{k-1}]], \\ &= \sum_{k=1}^{n} \omega_{k,n} \mathbb{E}[p_{k-1}^{-} + \int_{[0,1]^d} \mathbb{1}_{\{\mathbf{x} \in \mathbb{U}^- \cap supp(f_{k-1})\}} d\mathbf{x}]. \end{aligned}$$

Hence, if for all $k \geq 1$ and for all $\mathbf{x} \in \mathbb{U}^- \cap \mathbb{U}_{k-1}$, $f_{k-1}(\mathbf{x}) > 0$ then $\widetilde{p}_n$ is unbiased. Since this condition cannot be checked, it is sufficient to have $f_{k-1}(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathbb{U}_{k-1}$.

Remark 4.2 *Using classical results on importance sampling [102], $Var[\widetilde{p}_n]$ vanishes if for all $k \geq 1$,*

$$f_{k-1}(\mathbf{x}) = \frac{\mathbb{1}_{\{\mathbf{x} \in \mathbb{U}^- \cap \mathbb{U}_{k-1}\}}}{p - p_{k-1}^{-}}. \tag{4.7}$$

Of course this naive sampling scheme is not tractable as p is the unknown parameter. In a field testing point of view, the previous approach is not satisfactory. Indeed, it only provides a lower bound for p and a practitioner would expect an upper bound. In the next subsection, the previous approach is extend in order to avoid this drawback.

4.3 Sequential importance sampling-based estimation

In this section, a symmetrical construction of estimator (4.6) is considered. Set

$$\widehat{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p_{k-1}^{+} - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}} \right). \tag{4.8}$$

If f_{k-1} is the uniform distribution on $\mathbb{U}_{k-1}$, then estimators $\widetilde{p}_n$ and $\widehat{p}_n$ coincide. The analogous of (4.7) is stated in the following proposition.

Proposition 4.1 *If for all $k \geq 1$ and for all $\mathbf{x} \in \mathbb{U}_{k-1} \cap \mathbb{U}^+$, $f_{k-1}(\mathbf{x}) > 0$ then $\mathbb{E}[\widehat{p}_n] = p$. Moreover, if for all $k \geq 1$,*

$$f_{k-1}(\mathbf{x}) = \frac{\mathbb{1}_{\{\mathbf{x} \in \mathbb{U}^+ \cap \mathbb{U}_{k-1}\}}}{p_{k-1}^{+} - p}, \tag{4.9}$$

then $Var[\widehat{p}_n] = 0$.

As discussed before, as $\mathbb{U}^+$ is unknown the optimal density is useless. Nevertheless, the last proposition give us a practical guideline to perform a sampling density. For this, an estimate $\widehat{\mathbb{U}}^+_{k-1}$, based on response surface, of the unknown set $\mathbb{U}_{k-1} \cap \mathbb{U}^+$ is proposed (see Figure 4.2). Further, it is important for practitioner to work with unbiased estimate. An heuristic method aiming to decreasing the bias is provided. To check the unbiasedness condition stated in Proposition 4.1, the set $\widehat{\mathbb{U}}^+_{k-1}$ must be such that for all $k \geq 1, \widehat{\mathbb{U}}^+_{k-1} \subset \mathbb{U}^+ \cap \mathbb{U}_{k-1}$.

To overcome this obstacle, we proposed at step $k \geq 1$ to simulate $\mathbf{X}_k$ uniformly in $\widehat{\mathbb{U}}^+_{k-1}$ with probability $1 - \varepsilon_{k-1}$ and uniformly in $\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}$ with probability ε_{k-1}. In other words

$$\mathbf{X}_{k-1} \sim \begin{cases} \mathcal{U}(\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}) & \text{with probability } \varepsilon_{k-1}, \\ \mathcal{U}(\widehat{\mathbb{U}}^+_{k-1}) & \text{with probability } 1 - \varepsilon_{k-1}. \end{cases} \tag{4.10}$$

So that, conditionally to $\mathcal{F}_{k-1}$ and ε_{k-1}, the density of $\mathbf{X}_{k-1}$ is

$$f_{k-1}(\mathbf{x}) = \frac{\varepsilon_{k-1}}{\mu(\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1})} \mathbb{1}_{\{\mathbf{x} \in \mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}\}} + \frac{1 - \varepsilon_{k-1}}{\mu(\widehat{\mathbb{U}}^+_{k-1})} \mathbb{1}_{\{\mathbf{x} \in \widehat{\mathbb{U}}^+_{k-1}\}}. \tag{4.11}$$

Alleviating the notations, denote

$$\bar{p}_k = p^+_{k-1} - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}}. \tag{4.12}$$

Proposition 4.2 *The conditional variance of $\bar{p}_k$ is equal to*

$$Var(\bar{p}_k | \mathcal{F}_{k-1}) = \frac{a_{k-1}}{\varepsilon_{k-1}} + \frac{b_{k-1}}{1 - \varepsilon_{k-1}} - \left(p^+_{k-1} - p\right)^2,$$

where

$$a_{k-1} = \mu(\mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}) \mu\left(\mathbb{U}^+ \cap \mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}\right),$$
$$b_{k-1} = \mu(\widehat{\mathbb{U}}^+_{k-1}) \mu\left(\mathbb{U}^+ \cap \widehat{\mathbb{U}}^+_{k-1}\right).$$

The variance $Var(\bar{p}_k | \mathcal{F}_{k-1})$ goes to infinity while ε_{k-1} goes to 0 or 1. This can be explained by the role of ε_{k-1} that is, for instance, to ensure that $\bar{p}_k$ is unbiased. If ε_{k-1} goes to 0 or 1, the density f_{k-1} is not sufficiently counterbalanced by the uniform distribution.

Remark 4.3 *If the value of ε_{k-1} is fixed to 0 or 1, the estimator $\bar{p}_k$ can be biased. Nonetheless, its quadratic error can be expressed:*

$$\mathbb{E}[(\bar{p}_k - p)^2 | \mathcal{F}_{k-1}] = \begin{cases} (p^+_{k-1} - p)^2 - b_{k-1} - 2(p^+_{k-1} - p) \mu\left(\mathbb{U}^+ \cap \widehat{\mathbb{U}}^+_{k-1}\right) & \text{if } \varepsilon_{k-1} = 0, \\ (p^+_{k-1} - p)^2 - a_{k-1} - 2(p^+_{k-1} - p) \mu\left(\mathbb{U}^+ \cap \mathbb{U}_{k-1} \backslash \widehat{\mathbb{U}}^+_{k-1}\right) & \text{if } \varepsilon_{k-1} = 1. \end{cases}$$

How to built an estimate $\mathbb{U}_{k-1} \cap \mathbb{U}^+$? For example: **i)** the monotonic multi-layer perceptrons neural networks (MNN) published in [35]. **ii)** the combination of linear support vector machines (LSVM) described in Section 3.5 in the specific case where $\mathbb{U}^-$ or $\mathbb{U}^+$ is a convex set.

Under mild assumptions, both approaches provide consistent and comparable estimators of Γ (see Theorem 3.1). However, their computational cost can strongly differ: while MNN must be entirely calibrated from a static design of experiments (updated at each iteration), the

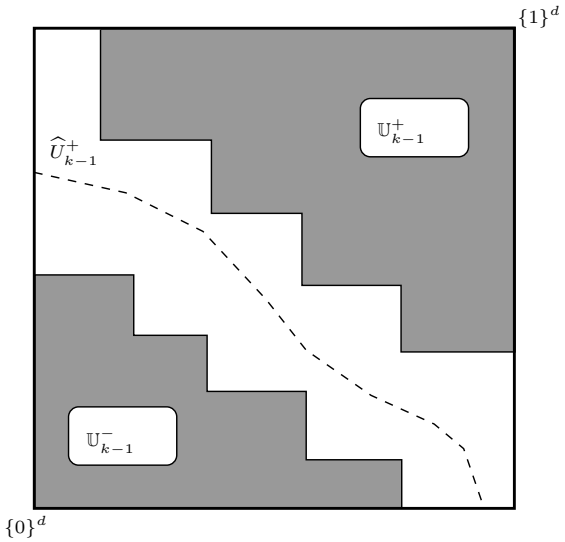


Figure 4.2: Illustration of the sequential sampling strategy in dimension 2. At step k, an estimator (dashed line) of Γ is proposed as well as an estimation of $\mathbb{U}_{k-1} \cap \mathbb{U}^+$ denoted $\widehat{\mathbb{U}}^+_{k-1}$.

calibration of LSVM is conducted in situ (dynamically). This last technique is more appropriate for our problem.

The asymptotic properties of $\widehat{p}_n$ are now discussed. Recall that for all $k \geq 1, \mathbf{X}_k$ is distributed as in (4.10) and denote f_{k-1} its probability density function. Then

$$\widehat{p}_n = \sum_{k=1}^{n} \omega_{k,n} \left(p^+_{k-1} - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}} \right). \tag{4.13}$$

Recall that $\widehat{p}_n$ is an unbiased estimator of p. The graal in the study of asymptotic properties of $\widehat{p}_n$ would be a result like

$$\frac{\widehat{p}_n - p}{\sqrt{Var\left[\widehat{p}_n\right]}} \underset{n \to +\infty}{\mathcal{L}} \mathcal{N}(0,1). \tag{4.14}$$

Following the construction in [62] (Chapter 3), $(\widehat{p}_n - p)/\sqrt{Var\left[\widehat{p}_n\right]}$ can be expressed as a martingale array. Let us start by the construction of the underlying martingale. Recall that

$$\bar{p}_k = p^+_{k-1} - \frac{1}{f_{k-1}(\mathbf{X}_k)} \mathbb{1}_{\{\mathbf{X}_k \in \mathbb{U}^+\}}.$$

Lemma 4.1 *Define for all* $k \geq 1$

$$M_k = \sum_{j=1}^{k} \tilde{\omega}_j (\bar{p}_j - p).$$

Then $\{M_k, \mathcal{F}_k, k \geq 1\}$ *is a zero-mean square-integrable martingale with*

$$Var(M_n) = \sum_{k=1}^{n} \tilde{\omega}_k^2 Var\left(\bar{p}_k\right).$$

Denote $M_{k,n} = \frac{M_k}{\sqrt{Var[M_n]}}$ and $\mathcal{F}_{k,n} = \mathcal{F}_k$, then $\{M_{k,n}, \mathcal{F}_{k,n},\ n \geq 1,\ k = 1, \dots, n\}$ is a martingale array. Hence, the graal (4.14) is equivalent to $M_{n,n} \underset{n\to+\infty}{\xrightarrow{\mathcal{L}}} \mathcal{N}(0,1)$.

Let $Z_{k,n} := M_{k,n} - M_{k-1,n}$. Corollary 3.1 in [62], states that if

(i) $\mathcal{F}_{k,n} \subset \mathcal{F}_{k,n+1}$ for $1 \leq k \leq n,\ n \geq 1$,

(ii) $\sum_{k=1}^{n} \mathbb{E}[Z_{k,n}^2 | \mathcal{F}_{k-1,n}] \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 1$,

(iii) for all $\varepsilon > 0$, $\sum_{k=1}^{n} \mathbb{E}[Z_{k,n}^2 \mathbb{1}_{\{|Z_{k,n}|>\varepsilon\}} | \mathcal{F}_{k-1,n}] \underset{n\to+\infty}{\xrightarrow{\mathbb{P}}} 0$ (Lindeberg condition),

then $M_{n,n} \xrightarrow{\mathcal{L}} \mathcal{N}(0,1)$.

Proposition 4.3 *If there exist $c > 0$ such that for all $k \geq 1, |Var[\bar{p}_k] - Var[\bar{p}_k|\mathcal{F}_{k-1}]| \leq c$ almost surely, and if*

$$\frac{\sum_{k=1}^{n} \tilde{\omega}_k^4}{\left(\sum_{k=1}^{n} \tilde{\omega}_k^2 Var[\bar{p}_k]\right)^2} \underset{n\to+\infty}{\longrightarrow} 0, \tag{4.15}$$

*then **(ii)** holds.*

Remark 4.4 *A necessary condition for (4.15) is*

$$\sum_{k=1}^{n} Var[\bar{p}_k] \underset{n\to+\infty}{\longrightarrow} +\infty. \tag{4.16}$$

The Azuma-Hoeffding inequality was used in the proof of Proposition 4.3. The quantity $Var[\bar{p}_k] - Var[\bar{p}_k|\mathcal{F}_{k-1}]$ must be bounded by a deterministic constant. In general, it is difficult to know exactly the variance of $\bar{p}_n$ and then Condition (4.15) is difficult to check. Nonetheless, using results in Section 3.3, the necessary condition (4.16) can be studied in a particular case.

Let $d = 1$ and assume that for all $k \geq 1, X_k$ is uniformly distributed on the non-dominated set. Then $\bar{p}_k = p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{g(X_k)\leq 0\}}$ and $Var[\bar{p}_k] = \mathbb{E}[(p_{k-1}^+ - p)(p - p_{k-1}^-)]$. Using Proposition 3.6, it comes $Var[\bar{p}_k] \leq (3/4)^k$ which does not verify (4.16).

Another example is discussed. Consider that f_{k-1} is given by (4.11) and recall that the variance of $\bar{p}_k$ goes to the infinity as ε_{k-1} goes to 0 or 1. Then, ε_{k-1} can be chosen such that the variance of $\bar{p}_k$ is bounded. If so, the upper bound is useful to verify the condition of Proposition 4.3 whereas the lower bound is helpful to check (4.16).

To conclude on this section, an estimator which converges too fast towards p implies it cannot be controlled by a central limit theorem.

4.4 Numerical experiments

In this section, the two estimators described below are compared on the following example described in [16]. Let $\mathbf{x} = (x^1, \dots, x^d) \in \mathbb{R}^d$ and denote

$$g(\mathbf{x}) = \frac{x^1}{\sum_{i=1}^d x^i}.$$

Let $\mathbf{U} = (U^1, \dots, U^d)$ be a random vector where $U^i \sim \Gamma(i+1, 1)$. Let

$$Z_d = g(\mathbf{U}) = U^1 / \sum_{i=1}^d U^i \sim \mathcal{B}eta(2, (d+1)(d+2)/2 - 3).$$

Further, let $q_{d,p}$ be the p-quantile of Z_d. The aim of this section is to estimation the probability given by $p = \mathbb{P}(g(\mathbf{U}) \leq q_{d,p})$. The function g is increasing according to its first argument and decreasing according to the other one. Denote F_i the cumulative distribution function of U^i. Set

$$\begin{aligned} X^1 &= F_1(U^1), \\ X^i &= 1 - F_i(U^i) \text{ for all } i = 2, \dots, d. \end{aligned}$$

So that, $\mathbf{X} = (X^1, \dots, X^d)$ is a random vector uniformly distributed on $[0,1]^d$. Hence

$$p = \mathbb{P}\left(g\left(F_1^{-1}\left(X^1\right), F_2^{-1}\left(1 - X^2\right), \dots, F_d^{-1}\left(1 - X^d\right)\right) - q_{d,p} \leq 0\right),$$

where F^{-1} is the inverse function of F.

The numerical experiments are conducted on two strategies of simulations. The first one is a uniform sampling on the non-dominated set $\mathbb{U}_{k-1}$. This strategy provides two estimators: the maximum likelihood estimator (MLE) built in [16] and the estimator (4.6) with constant weights $\omega_{k,n} = 1/n$ for all $k = 1, \dots, n$. The second strategy uses the optimal density (4.9): at step k, a random vector is uniformly distributed on $\mathbb{U}_{k-1} \cap \mathbb{U}^+$. The estimator obtained is almost surely equal to p, and of course only the upper bound is discussed.

Denote for all $n \geq 1$

$$L_n(r) := L_n(\mathbf{x}_1, \dots, \mathbf{x}_n | r) = \prod_{k=1}^n \left(\frac{r - p_{k-1}^-}{p_{k-1}^+ - p_{k-1}^-}\right)^{\mathbb{1}_{\{\mathbf{x}_k \in \mathbb{U}^-\}}} \left(\frac{p_{k-1}^+ - r}{p_{k-1}^+ - p_{k-1}^-}\right)^{1 - \mathbb{1}_{\{\mathbf{x}_k \in \mathbb{U}^-\}}}.$$

The two estimators are

$$\begin{aligned} \hat{p}_n^{MLE} &:= \underset{r \in]p_{n-1}^-, p_{n-1}^+[}{\arg\max} \; L_n(r), \\ \widehat{p}_n &:= \frac{1}{n} \sum_{k=1}^n p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-) \mathbb{1}_{\{\mathbf{x}_k \in \mathbb{U}^-\}}. \end{aligned} \tag{4.17}$$

One of the main objective of this numerical study is to compare both the role of the dimension d and the magnitude of p on these two estimators. The comparison is conducted on the following couples (d, p):

$$\begin{aligned} &\left(3, 10^{-3}\right), \left(4, 10^{-3}\right), \left(5, 10^{-3}\right), \\ &\left(3, 10^{-5}\right), \left(4, 10^{-5}\right), \left(5, 10^{-5}\right). \end{aligned}$$

The quadratic errors and the bias of the estimators as well as the quadratic error of the upper bound are computed. The relative quadratic error and the relative bias are defined as follow

$$\frac{\mathbb{E}[(c_n - c)^2]}{c}, \frac{\mathbb{E}[c_n - c]}{c},$$

where c_n is one of the examined quantities and c the true quantity.

Figures 4.3 and 4.4 provide the results obtained with the two estimators and different values of p. From left to right the dimension is respectively equal to $3, 4$ and 5.

For $p = 10^{-3}$, the two estimators (represented in blue plain line for the MLE and in red dashed line for the other one) seems to have equivalent performance in terms of bias and quadratic error. Their bias seems to increase with the dimension.

Nonetheless, the properties of the MLE are degraded when p becomes very small. As studied in Chapter 3, this can be explained by the increasing distance of the deterministic bounds with p when the dimension increases. But $\widehat{p}_n$ seems not to suffer from the dimension or the magnitude of p.

The rate of convergence of the upper bound is compared on Figure 4.5 with a uniform sampling strategy and using the optimal density (4.9). Even if this ideal importance density cannot be used in practice to estimate p, it can be used to make such a comparison. A simple rejection method on the non-dominated set is used. The considered sample size is the number of non-rejected simulations. The upper bound obtained from a uniform (resp. optimal) sampling is represented in a blue plain line (resp. green dotted line). The upper bound appears to be slightly influenced by the strategy of simulation. It must be noticed that simulating uniformly on the non-dominated set is equivalent to choose $\widehat{\mathbb{U}}^+_{k-1} = \mathbb{U}^+ \cap \mathbb{U}_{k-1}$ and $\varepsilon_{k-1} = (p^+_{k-1} - p)/(p^+_{k-1} - p^-_{k-1})$. In this example, the probability $\mathbb{P}(\mathbf{X}_k \in \mathbb{U}_{k-1} \cap \mathbb{U}^+) = \mathbb{E}[\varepsilon_{k-1}]$ is close to 1 (see Figure 4.5). These two remarks allow to say that the uniform distribution is close the optimal one.

To conclude on this example, it seems that the most tractable method in practice is the uniform simulation method provided in [16]. It does not require to make an estimation of the limit state or g and allows to build an unbiased estimator of p. Moreover, first numerical studies show that it is difficult to get both an unbiased estimator and a significant reduction of the deterministic upper bound. The use of the optimal density does not improve significantly the convergence of the upper bound on this example.

4.5 Conclusion

The aim of this chapter was twofold. First, to obtain an unbiased estimator of p with a lower quadratic error than the MLE proposed in [16]. Second, to provide a sharper deterministic upper bound. Despite a complicated construction, the use of an optimal density does not provide a significantly improvement.

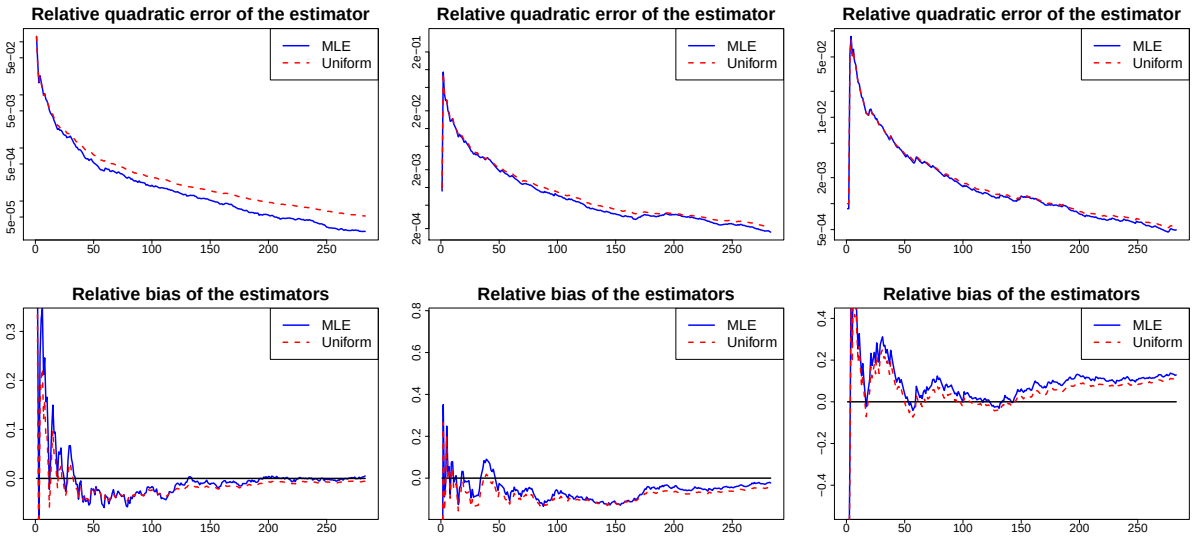


Figure 4.3: Toy example for $p = 10^{-3}$ in function of $n = 1, \ldots, 300$. For the figure in left and in the middle, the estimation obtained by the optimal density is not represented since it is almost surely equal to p. **From left to right:** $d = 3, 4, 5$. **First line:** Comparison of the quadratic error obtained with the two different estimators. **Second line:** Comparison of the bias obtained with the two different estimators. Results have been averaged on 210 independent experiments.

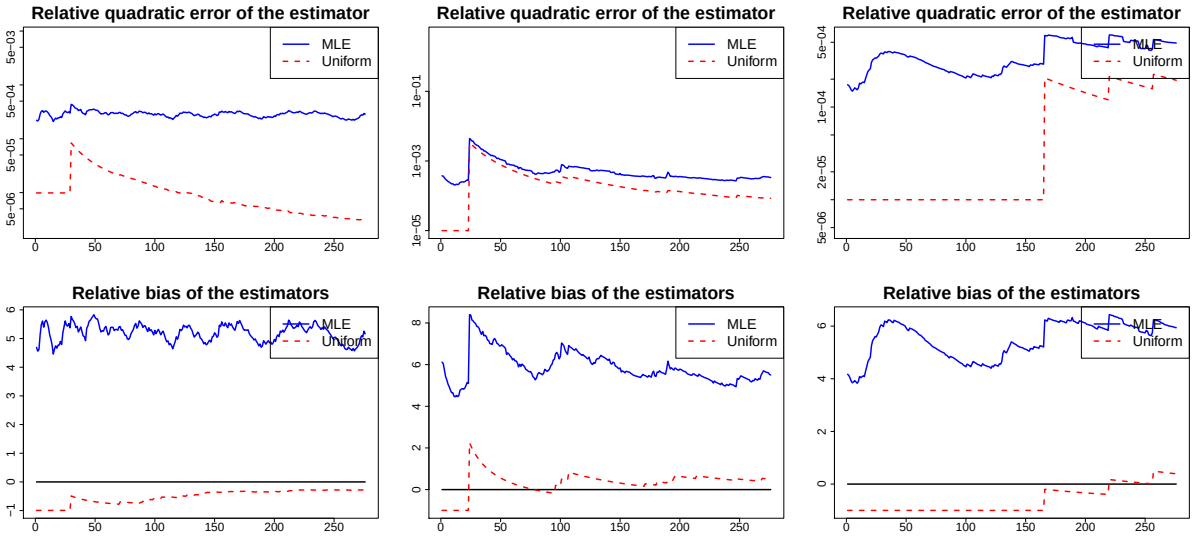


Figure 4.4: Toy example for $p = 10^{-5}$ in function of $n = 1, \ldots, 300$. For the figure in left and in the middle, the estimation obtained by the optimal density is not represented since it is almost surely equal to p. **From left to right:** $d = 3, 4, 5$. **First line:** Comparison of the quadratic error obtained with the two different estimators. **Second line:** Comparison of the bias obtained with the two different estimators. Results have been averaged on 132 independent experiments.

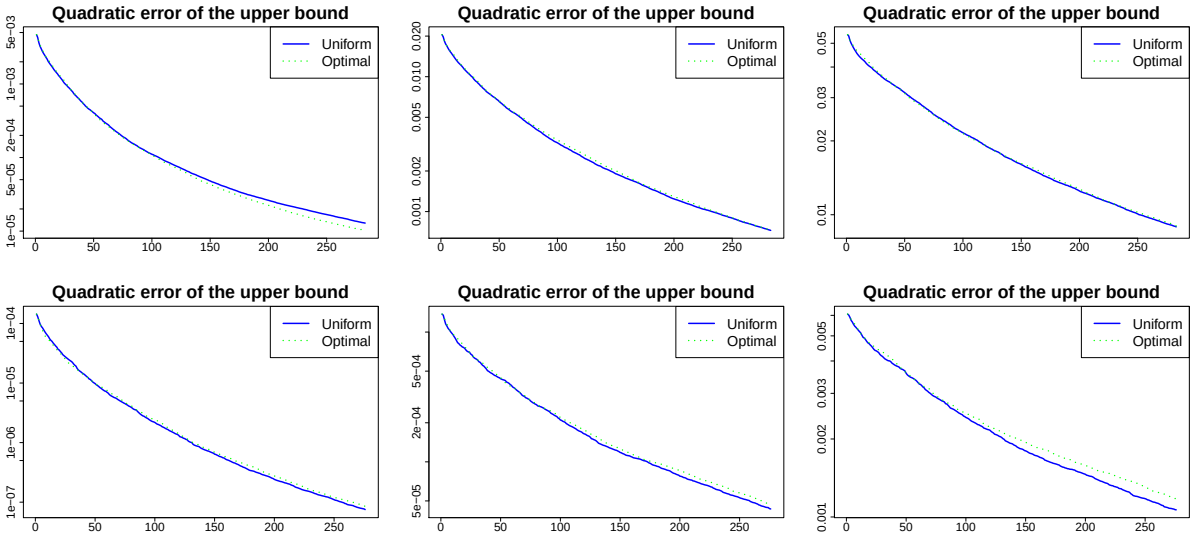


Figure 4.5: Toy example for $p \in \{10^{-3}, 10^{-5}\}$ in function of $n = 1, \ldots, 300$. Comparison of the quadratic error of the upper bound for the two different strategies of simulation. **From left to right:** $d = 3, 4, 5$. **From up to down:** $p = 10^{-3}, 10^{-5}$. Results have been averaged on 210 (resp. 132) independent experiments for $p = 10^{-3}$ (resp. $p = 10^{-5}$).

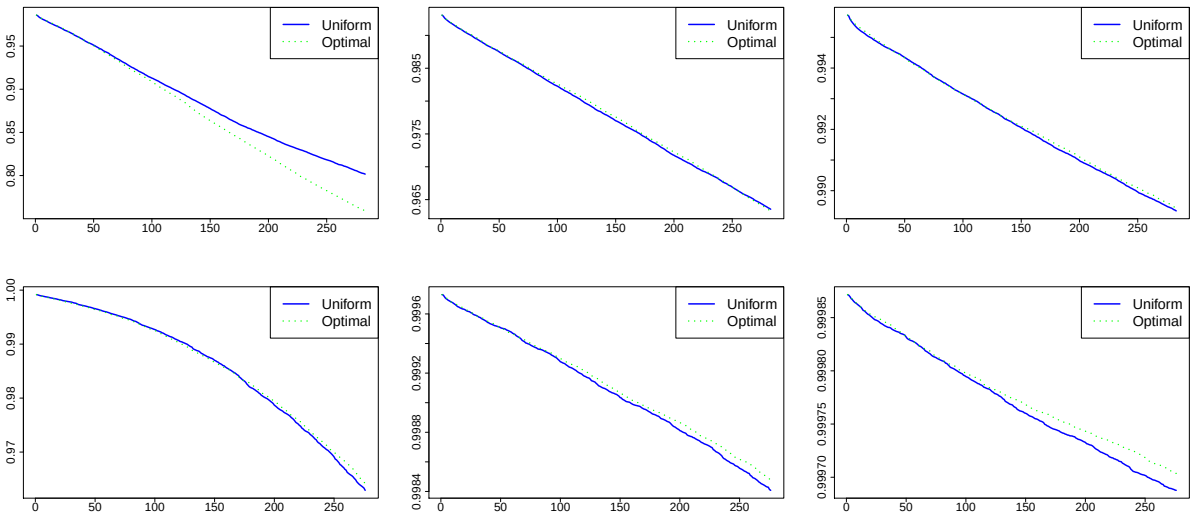


Figure 4.6: Toy example for $p \in \{10^{-3}, 10^{-5}\}$ in function of $n = 1, \ldots, 200$. Representation of $\mathbb{P}(\mathbf{X}_n \in \mathbb{U}_{n-1} \cap \mathbb{U}^+)$ with $\mathbf{X}_n$ uniformly distributed on the non-dominated set $\mathbb{U}_{n-1}$. **From left to right:** $d = 3, 4, 5$. **From up to down:** $p = 10^{-3}, 10^{-5}$. Results have been averaged on 210 (resp. 132) independent experiments for $p = 10^{-3}$ (resp. $p = 10^{-5}$).

4.6 Proofs

Proof of Proposition 4.2. Since $\bar{p}_k$ is unbiased, it remains to compute $\mathbb{E}[\bar{p}_k^2|\mathcal{F}_{k-1}]$.

$$\begin{aligned}\mathbb{E}[\bar{p}_k^2|\mathcal{F}_{k-1}] &= (p_{k-1}^+)^2 - 2p_{k-1}^+\mathbb{E}\left[\frac{1}{f_{k-1}(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right] + \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right],\\ &= (p_{k-1}^+)^2 - 2p_{k-1}^+(p_{k-1}^+ - p) + \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right],\\ &= 2pp_{k-1}^+ - (p_{k-1}^+)^2 + \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right].\end{aligned}$$

It comes

$$\begin{aligned}Var(\bar{p}_k|\mathcal{F}_{k-1}) &= \mathbb{E}\left[\bar{p}_k^2|\mathcal{F}_{k-1}\right] - \mathbb{E}^2\left[\bar{p}_k|\mathcal{F}_{k-1}\right],\\ &= 2pp_{k-1}^+ - (p_{k-1}^+)^2 + \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right] - p^2,\\ &= \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right] - (p_{k-1}^+ - p)^2.\end{aligned}$$

But

$$\begin{aligned}\mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}|\mathcal{F}_{k-1}\right] &= \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}^+\}}\mathbb{1}_{\{\mathbf{X}_k\in\mathbb{U}_{k-1}\backslash\widehat{\mathbb{U}}_{k-1}^+\}}|\mathcal{F}_{k-1}\right]\\ &\quad + \mathbb{E}\left[\frac{1}{f_{k-1}^2(\mathbf{X}_k)}\mathbb{1}_{\{\mathbf{X}_k\in\widehat{\mathbb{U}}_{k-1}^+\}}|\mathcal{F}_{k-1}\right],\\ &= \int_{\mathbb{U}^+\cap\mathbb{U}_{k-1}\backslash\widehat{\mathbb{U}}_{k-1}^+}\frac{\mu(\mathbb{U}_{k-1}\backslash\widehat{\mathbb{U}}_{k-1}^+)}{\varepsilon_{k-1}}d\mathbf{x} + \int_{\mathbb{U}^+\cap\widehat{\mathbb{U}}_{k-1}^+}\frac{\mu(\widehat{\mathbb{U}}_{k-1}^+)}{1-\varepsilon_{k-1}}d\mathbf{x},\\ &= \frac{\mu(\mathbb{U}_{k-1}\backslash\widehat{\mathbb{U}}_{k-1}^+)}{\varepsilon_{k-1}}\mu\left(\mathbb{U}^+\cap\mathbb{U}_{k-1}\backslash\widehat{\mathbb{U}}_{k-1}^+\right)\\ &\quad + \frac{\mu(\widehat{\mathbb{U}}_{k-1}^+)}{1-\varepsilon_{k-1}}\mu\left(\mathbb{U}^+\cap\widehat{\mathbb{U}}_{k-1}^+\right),\end{aligned}$$

which conclude the proof. □

Proof 4.1 *(Lemma 4.1.) Let $k \geq 1$ and define $M_k = \sum_{j=1}^{k}\tilde{\omega}_j(\bar{p}_j - p)$. It comes*

$$\begin{aligned}\mathbb{E}[M_k] &= \sum_{j=1}^{k}\tilde{\omega}_j\mathbb{E}[\bar{p}_j - p],\\ &= 0,\end{aligned}$$

since $\bar{p}_j$ is unbiased estimator of p.

$$\begin{aligned}\mathbb{E}[M_k|\mathcal{F}_{k-1}] &= \sum_{j=1}^{k-1}\tilde{\omega}_j(\bar{p}_j - p) + \tilde{\omega}_k\mathbb{E}[\bar{p}_k - p|\mathcal{F}_{k-1}],\\ &= M_{k-1},\end{aligned}$$

since $\bar{p}_k$ is an unbiased estimator of p. Moreover,

$$M_k^2 = \tilde{\omega}_k^2(\bar{p}_j - p)^2 + 2\sum_{\substack{i,j=1\\ i<j}}^{k-1} \tilde{\omega}_i\tilde{\omega}_j(\bar{p}_i - p)(\bar{p}_j - p),$$

then

$$\begin{aligned}\mathbb{E}[M_k^2] &= \sum_{j=1}^{k} \tilde{\omega}_j^2\mathbb{E}[(\bar{p}_j - p)^2] + 2\sum_{\substack{i,j=1\\ i<j}}^{k-1} \tilde{\omega}_i\tilde{\omega}_j\mathbb{E}[(\bar{p}_i - p)(\bar{p}_j - p)],\\ &= \sum_{j=1}^{k} \tilde{\omega}_j^2\mathbb{E}[(\bar{p}_j - p)^2].\end{aligned}$$

Indeed, $\mathbb{E}[(\bar{p}_i - p)(\bar{p}_j - p)] = \mathbb{E}[(\bar{p}_i - p)\mathbb{E}[\bar{p}_j - p|\mathcal{F}_{j-1,n}]] = 0$. Finally, it comes

$$\mathbb{E}[M_k^2] = \sum_{j=1}^{k} \tilde{\omega}_j^2 Var\left[\bar{p}_j\right] < +\infty,$$

and then $\{M_k, \mathcal{F}_k, k \geq 1\}$ is a zero-mean square-inegrable martingale. □

Proof of Proposition 4.3. Denote

$$\begin{aligned}Z_{k,n} &= M_{k,n} - M_{k-1,n},\\ &= \frac{\tilde{\omega}_k(\bar{p}_k - p)}{\left(\sum_{j=1}^{n} \tilde{\omega}_j^2 Var\left(\bar{p}_j\right)\right)^{1/2}},\end{aligned}$$

then

$$Z_{k,n}^2 = \frac{\tilde{\omega}_k^2(\bar{p}_k - p)^2}{\sum_{j=1}^{n} \tilde{\omega}_j^2 Var\left(\bar{p}_j\right)}, \tag{4.18}$$

and

$$\sum_{k=1}^{n} \mathbb{E}[Z_{k,n}^2] = 1. \tag{4.19}$$

Let $\varepsilon > 0$, using (4.19) then (4.18), it comes

$$\begin{aligned}\mathbb{P}\left(\left|\sum_{k=1}^{n} \mathbb{E}[Z_{k,n}^2|\mathcal{F}_{k-1,n}] - 1\right| > \varepsilon\right) &= \mathbb{P}\left(\left|\sum_{k=1}^{n} \left(\mathbb{E}[Z_{k,n}^2|\mathcal{F}_{k-1,n}] - \mathbb{E}[Z_{k,n}^2]\right)\right| > \varepsilon\right),\\ &= \mathbb{P}\left(\frac{\left|\sum_{k=1}^{n} \tilde{\omega}_k^2\left(Var\left[\bar{p}_k\right|\mathcal{F}_{k-1,n}] - Var\left[\bar{p}_k\right]\right)\right|}{\sum_{k=1}^{n} \tilde{\omega}_k^2 Var\left(\bar{p}_k\right)} > \varepsilon\right).\end{aligned}$$

Define by

$$A_n = \sum_{k=1}^{n} \tilde{\omega}_k^2 \left(Var\left[\bar{p}_k \middle| \mathcal{F}_{k-1,n}\right] - Var\left[\bar{p}_k\right] \right),$$

a zero-mean martingale. From hypothesis, it comes

$$\begin{aligned} |A_n - A_{n-1}| &= \tilde{\omega}_n^2 |Var\left[\bar{p}_n \middle| \mathcal{F}_{n-1,n}\right] - Var\left[\bar{p}_n\right]|, \\ &\leq c\tilde{\omega}_n^2. \end{aligned}$$

Azuma-Hoeffding inequality states that if there exists a sequence of non negative real values $(c_n)_{n\geq 1}$ such that $\mathbb{P}(|A_n - A_{n-1}| \leq c_n) = 1$ for all $n \geq 1$, then for all $\lambda > 0$

$$\mathbb{P}(|A_n| \geq \lambda) \leq 2\exp\left(-2\lambda^2 / \sum_{k=1}^{n} c_k^2 \right).$$

Let $\varepsilon > 0$ and set $\lambda = \sum_{k=1}^{n} \tilde{\omega}_k^2 Var\left(\bar{p}_k\right) \varepsilon$. Then

$$\mathbb{P}\left(\left| \sum_{k=1}^{n} \mathbb{E}[Z_{k,n}^2 | \mathcal{F}_{k-1,n}] - 1 \right| > \varepsilon \right) \leq 2\exp\left(-2\varepsilon^2 \left(\sum_{k=1}^{n} \tilde{\omega}_k^2 Var\left(\bar{p}_k\right) \right)^2 / \sum_{k=1}^{n} c^2 \tilde{\omega}_k^4 \right).$$

Now, let us study the necessary condition:

$$\begin{aligned} \frac{\sum_{k=1}^{n} \tilde{\omega}_k^4}{\left(\sum_{k=1}^{n} \tilde{\omega}_k^2 Var\left[\bar{p}_k\right] \right)^2} &\geq \frac{\sum_{k=1}^{n} \tilde{\omega}_k^4}{\left(\max_{1\leq k\leq n} \tilde{\omega}_k^2 \right)^2 \left(\sum_{k=1}^{n} Var\left[\bar{p}_k\right] \right)^2}, \\ &\geq \frac{\sum_{k=1}^{n} \tilde{\omega}_k^4}{\sum_{k=1}^{n} \tilde{\omega}_k^4 \left(\sum_{k=1}^{n} Var\left[\bar{p}_k\right] \right)^2}, \\ &\geq \frac{1}{\left(\sum_{k=1}^{n} Var\left[\bar{p}_k\right] \right)^2}. \end{aligned}$$

□

Chapter 5

Quantile estimation under monotonicity constraint

Résumé. Ce chapitre traite de l'estimation d'un quantile d'ordre p de la variable aléatoire $g(\mathbf{X})$ définie par $q = \inf\{t \in \mathbb{R},\ \mathbb{P}(g(\mathbf{X}) \leq t) > p\}$. On reprend les hypothèses du chapitre précédent: g est une fonction de type boîte noire, coûteuse en temps de calcul et globalement monotone. À partir d'un plan d'expérience donné vérifiant une contrainte géométrique, des bornes déterministes de ce quantile sont obtenues. Comme pour l'estimation de probabilité, l'utilisation de ces bornes permet de mener une exploration séquentielle de l'espace d'entrée du code numérique et de proposer un estimateur consistant de q.

Abstract. This chapter deals with the estimation of a p-quantile of the random variable $g(\mathbf{X})$ defined by $q = \inf\{t \in \mathbb{R},\ \mathbb{P}(g(\mathbf{X}) \leq t) > p\}$. Keeping the assumptions of the previous chapter: g is a black-box function, time-consuming and globally monotone. From a design of numerical experiments that satisfy a geometric constraint, two deterministic bounds of this quantile are obtained. As for probability estimation, the use of these bounds allows to make a sequential exploration of the input space and provide a consistent estimator of q.

5.1 Introduction

As before, g is a measurable globally increasing function on $[0,1]^d$. Without loss of generality it is consider that $\mathbf{X}$ is uniformly distributed on $[0,1]^d$. Denote F the cumulative distribution function of $g(\mathbf{X})$, and let $p \in]0,1[$. The p-quantile of $g(\mathbf{X})$ is defined by

$$q = \inf\{t \in \mathbb{R},\ F(t) \geq p\}. \tag{5.1}$$

Most non-intrusive methods of quantile estimation are based on the computation of the inverse of the cumulative distribution function F [36, 58, 85]. A general estimator of q is defined by

$$\widehat{q} = \inf\{t \in \mathbb{R},\ \widehat{F}(t) \geq p\}, \tag{5.2}$$

where $\widehat{F}$ is an estimator of F. Monte Carlo methods require to produce a large sample and to use the empirical cumulative distribution function built from this sample. Variance reduction techniques are usually applied to estimate the cumulative distribution function. One of them is importance sampling method [58]. In general, finding the importance sampling distribution can be difficult. Taking into account of information given by the simulations, adaptive methods

must be used [52, 85]. The splitting method introduced in [3] and developed in [61] consists in simulating closer and closer to the limit surface $\Gamma = \{\mathbf{x} \in \mathbb{R}^d,\ g(\mathbf{x}) = q\}$. Another way to limit the number of calls of the numerical code is to use a meta-modelling method in order to mimic the output [20].

The main advantages of monotonicity is to provide deterministic information on q and to build a subset in the input space where the sign of $g(.) - q$ is known. Following [16] and Chapter 4, a local estimator of $F(q)$, useful for quantile estimation, is provided.

The chapter is structured as follows. Section 5.3 provides an initialisation step to get two bounds for q requiring only two runs of g. Besides, a method to bound a quantile from a set of points is proposed. In Section 5.4 an estimator of $F(q)$ is built. It has the same form as (4.6), then the quantile is estimated as described in (5.2). Finally, the method is tested on a numerical example and compared with different classical quantile estimators in Section 5.5. Proofs are stated in Section 5.7.

5.2 State of the art for rare quantile estimation

In this section some classical methods of quantile estimation is discussed.

The empirical quantile estimator is built from a iid realisations. From this sample an unbiased estimator $\widehat{F}$ of F is built and then q is estimated as in (5.2). Let $(Z_k)_{k\geq 1}$ be a sequence of iid random variables with common cumulative distribution function F and probability density function f such that $f(q) > 0$. Let $t \in \mathbb{R}$, the cumulative distribution F of Z can be estimated by the empirical cumulative distribution function

$$\widehat{F}_n(t) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}_{\{Z_k \leq t\}}.$$

Obviously $\mathbb{E}[\widehat{F}_n(t)] = F(t)$. Let $Z_{(1)}, \ldots, Z_{(n)}$ be the order statistic of $Z_1, \ldots, Z_n$. That is $Z_{(1)} \leq \cdots \leq Z_{(n)}$. Since $\lfloor np \rfloor / n \leq p \leq (\lfloor np \rfloor + 1)/n$ and using that for all $k = 1, \ldots, n$, $\widehat{F}_n(Z_{(k)}) = k/n$, it comes

$$\widehat{F}_n(Z_{(\lfloor np \rfloor)}) \leq p \leq \widehat{F}_n(Z_{(\lfloor np \rfloor + 1)}).$$

The empirical estimator is given by

$$\widehat{q}_n^{emp} = Z_{(\lfloor np \rfloor + 1)}.$$

A central limit theorem on $\widehat{q}_n^{emp}$ can be found in [1]:

$$\sqrt{n}(\widehat{q}_n^{emp} - q) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(0, \frac{p(1-p)}{f(q)^2}\right).$$

Equivalent results have been proposed in [58] for quantile estimation using importance sampling. Let $(Y_k)_{k\geq 1}$ be a sequence of iid random variables distributed according to the density f_Y. For all $t \in \mathbb{R}$

$$\widetilde{F}_n(t) = \frac{1}{n}\sum_{k=1}^{n} \frac{f(Y_k)}{f_Y(Y_k)} \mathbb{1}_{\{Y_k \leq t\}},$$

is an unbiased estimator of $F(t)$. As in the standard case, a central limit theorem can be obtained (see Theorem 1 in [58]):

$$\sqrt{n}(\widetilde{q}_n - q) \xrightarrow[n\to+\infty]{\mathcal{L}} \sigma\mathcal{N}(0,1),$$

with $\sigma = \left(\mathbb{E}\left[\frac{f^2(Y_k)}{f_Y^2(Y_k)}\mathbb{1}_{\{Y_k\leq t\}}\right] - p^2\right)/f(q)^2$.

Quantile regression [73] is a variational way to built an estimate of q. Let

$$\rho_p(u) = |u(p - \mathbb{1}_{\{u<0\}})|,$$

a loss function. If q is the p-quantile, then q is also the minimiser of $\min_{\mathbf{u}} \mathbb{E}[\rho_p(Z-u)]$. Since the law of Z is unknown, the quantile regression is estimated from an iid sample $Z_1, \ldots, Z_n$. Finally, q is estimated by

$$\widehat{q}_n^{reg} = \arg\min_{u\in\mathbb{R}} \sum_{i=1}^{n} [\rho_p(Z_i - u)].$$

An alternative way to construct an estimator of q is possible using a particle algorithm. Indeed, an estimator $\widehat{q}_N^{last}$ is provided in [61] and verified the central limit theorem (5.3) (see Proposition 3 in [61]).

$$\sqrt{N}(\widehat{q}_N^{last} - q) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(0, \frac{-p^2\log p}{F'(q)^2}\right). \tag{5.3}$$

The non-parametric adaptive importance sampling (NAIS) is a sequential importance sampling developed in [85]. In general, such construction aims to produce an importance density close to the optimal one using a non-parametric sequential tools.

5.3 Quantile deterministic bounding

In this section, the monotonicity of g is exploited to get non-trivial bounds for q. As usual, assume 4.1, 4.2 and

Assumption 5.1 *Assume that* $\Gamma := \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\}$ *is simply connex and* $\mu(\Gamma) = 0$.

Denote

$$\mathbb{U}^- := \left\{\mathbf{x} \in [0,1]^d,\ g(\mathbf{u}) \leq q\right\},$$
$$\mathbb{U}^+ := \left\{\mathbf{x} \in [0,1]^d,\ g(\mathbf{u}) > q\right\}.$$

Since $g(.)$ is not transformed in $g(.) - q$, we use the same notations as in the previous chapters.

To bound the quantile q, we will used geometric arguments. The following definition is useful to alleviate the notation.

Definition 5.1 *Let* $A \subset [0,1]^d$. *Define*

$$\mathbb{V}^-(A) := \bigcup_{\mathbf{x}\in A} \{\mathbf{u} \in [0,1]^d :\ \mathbf{u} \preceq \mathbf{x}\}, \tag{5.4}$$
$$\mathbb{V}^+(A) := \bigcup_{\mathbf{x}\in A} \{\mathbf{u} \in [0,1]^d :\ \mathbf{u} \succeq \mathbf{x}\}. \tag{5.5}$$

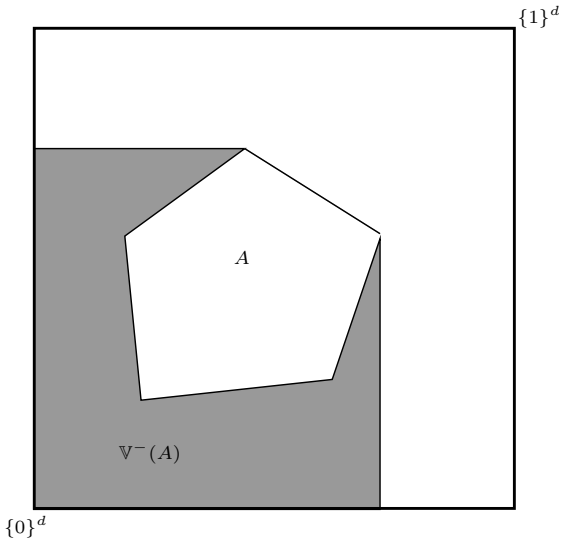


Figure 5.1: Illustration of a dominated set for $d = 2$. Let A be a subset of $[0,1]^2$. $\mathbb{V}^-(A)$ is the union of A with the set represented in gray.

It must be noticed that Equations (5.4) and (5.5) are slightly different from Equations (4.2) and (4.3) since they do not depend on the set $\mathbb{U}^-$. See Figure 5.1 for an illustration of $\mathbb{V}^-(A)$.

Definition 5.2 *Let $\alpha \in]0,1[$ and S be a set in $[0,1]^d$. The set S is said α-monotonic if for all $\mathbf{u}, \mathbf{v} \in S$, $\mathbf{u}$ is not strictly dominated by $\mathbf{v}$ and if $\mu(\mathbb{V}^-(S)) = \alpha$.*

Remark 5.1 *A consequence of the monotonicity of g is that the set $\{\mathbf{x} \in [0,1]^d,\ g(\mathbf{x}) = \alpha\}$ is $F(\alpha)$-monotonic. In particular, the limit surface Γ is p-monotonic.*

5.3.1 Initialisation

Without any call of g, the initialisation step provides two sets where the sign of $g(.) - q$ is known. The dichotomy procedure described in Section 2.3.2 cannot be applied since it depends on the sign of $g(.) - q$. The main ideas of this section are based on the following proposition.

Proposition 5.1 *Let S_p be a p-monotonic set then*

$$\min_{\mathbf{x} \in S_p} g(\mathbf{x}) \leq q \leq \max_{\mathbf{x} \in S_p} g(\mathbf{x}). \tag{5.6}$$

A monotonic set verifying properties of Proposition 5.1 can be difficult to build in practice. For instance, let us start this initialisation step with only one point $\mathbf{x} = (x^1, \ldots, x^d) \in [0,1]^d$. Since g is globally increasing, its maximum on $\mathbb{V}^-(\mathbf{x})$ is reached in $\mathbf{x}$ (see Figure 5.2 left). Since $\mu(\mathbb{V}^-(\mathbf{x})) = x^1 \cdots x^d$, choosing $x^i = p^{1/d}$ for all $i = 1, \ldots, d$ verify the constraint $\mu(\mathbb{V}^-(\mathbf{x})) = p$, and from Proposition 5.1, $q \leq g(\mathbf{x})$.

Using a symmetrical approach a lower bound can be easily obtained. Let $\mathbf{x} = (x^1, \ldots, x^d) \in [0,1]^d$, if $\mu(\mathbb{V}^+(\mathbf{x})) \geq 1 - p$ then the minimum of g on $\mathbb{V}^+(\mathbf{x})$ is reached on $\mathbf{x}$ and $g(\mathbf{x}) \leq q$. Since $\mu(\mathbb{V}^+(\mathbf{x})) = (1 - x^1) \cdots (1 - x^d)$, it is sufficient to choose $x^i = 1 - (1 - p)^{1/d}$ (see Figure

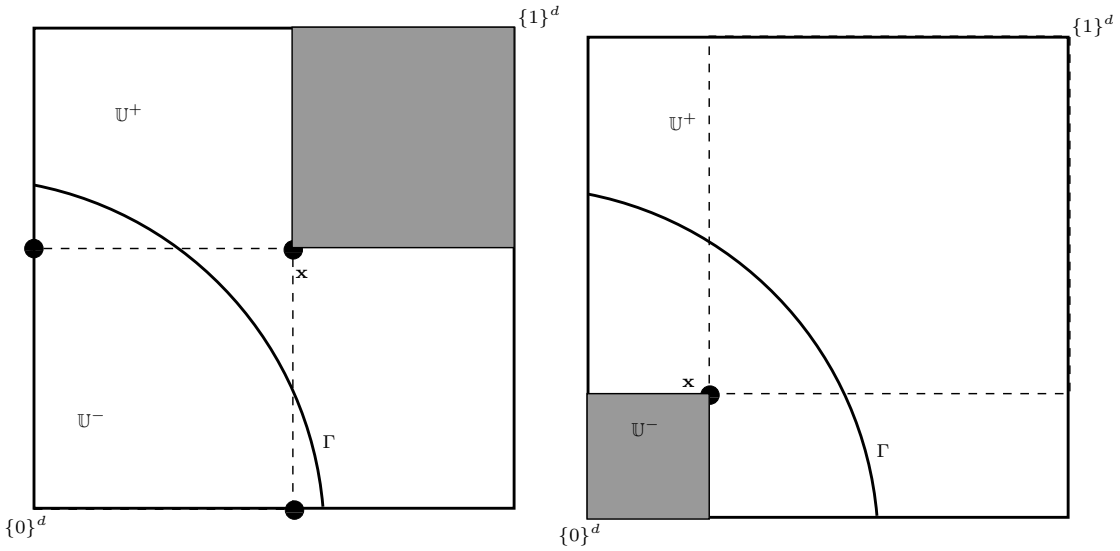


Figure 5.2: Illustration of the initialisation step with one point $\mathbf{x} = (x^1, \dots, x^d)$ for $d = 2$. Left (resp. Right): initialisation for the upper (resp. lower) bound. The point $\mathbf{x}$ is chosen such that for all $i = 1 \dots, d,\ x^i = p^{1/d}$ (resp. $x^i = 1 - (1-p)^{1/d}$). The dashed line represents the frontier of $\mathbb{V}^-(\mathbf{x})$ (resp. $\mathbb{V}^+(\mathbf{x})$) and its Lebesgue measure is equal to p (resp. $1-p$). For all $\mathbf{u}$ in the gray square $g(\mathbf{u}) \geq q$ (resp. $g(\mathbf{u}) \leq q$).

5.2 right). Finally, let

$$\mathbf{x}^- = \left(1 - (1-p)^{1/d}, \dots, 1 - (1-p)^{1/d}\right), \tag{5.7}$$

$$\mathbf{x}^+ = (p^{1/d}, \dots, p^{1/d}), \tag{5.8}$$

then $g(\mathbf{x}^-) \leq q \leq g(\mathbf{x}^+)$.

Remark 5.2 *For $d = 1$, it comes $x^- = x^+ = p$, then $g(x^-) = g(x^+) = g(p) = q$. In the remainder of the chapter it is always assumed that $d \geq 2$.*

The intuition behind this construction comes from Proposition 5.2 below. This proposition states that the extremal elements of the convex set of a globally decreasing function is the set of indicator functions.

Proposition 5.2 *Denote $C = \{f : [0,1]^{d-1} \to [0,1], f$ is globally decreasing$\}$. The set of extremal elements of C is*

$$\left\{ f : [0,1]^{d-1} \to [0,1], f(\mathbf{x}) = \mathbb{1}_{\left\{\mathbf{x} \in \cup_{k \geq 1} P_k\right\}}, P_k \in \mathcal{P} \right\},$$

where $\mathcal{P}$ is the set of hyper-rectangles in $[0,1]^{d-1}$ containing $\mathbf{0}$.

As said above, the volume constraint can be difficult to verify. Nonetheless this constraint can be relaxed as explained in Proposition 5.3.

Proposition 5.3 *For all $\alpha \in]0,1[$ assume that S_α is an α-monotonic set. Let $(p^-, p^+) \in [0, p[\times [p, 1[$, then*

$$\min_{\mathbf{x} \in S_{p^-}} g(\mathbf{x}) \leq q \leq \max_{\mathbf{x} \in S_{p^+}} g(\mathbf{x}). \tag{5.9}$$

The main advantage of Proposition 5.3 is that p^- and p^+ can be unknown. Indeed, it is sufficient to build a monotonic set S such that $\mu(\mathbb{V}^-(S)) \geq p$ or $\mu(\mathbb{V}^-(S)) \leq p$. Proposition 5.4 (see also Figure 5.3) provides two subsets of $[0,1]^d$ such that the value of g is either lower than q or greater than q. This construction depends only on the monotonicity and p. It does not require any evaluation by the numerical code.

Proposition 5.4 *Let $p \in]0,1[$ and $d \geq 2$. Denote*

$$\mathbb{W}^-(p) = \left\{ \mathbf{u} = (u^1, \ldots, u^d) \in [0,1]^d, \prod_{i=1}^{d} (1-u^i) \geq 1-p \right\},$$
$$\mathbb{W}^+(p) = \left\{ \mathbf{u} = (u^1, \ldots, u^d) \in [0,1]^d, \prod_{i=1}^{d} u^i \geq p \right\}.$$

For all $(\mathbf{u}, \mathbf{v}) \in \mathbb{W}^-(p) \times \mathbb{W}^+(p)$ it comes

$$g(\mathbf{u}) \leq q \leq g(\mathbf{v}).$$

Denoting $\mathbb{W}(p) = [0,1]^d \backslash (\mathbb{W}^-(p) \cup \mathbb{W}^+(p))$, then

$$\mu(\mathbb{W}(p)) = (1-p) \sum_{k=0}^{d-1} \left[\frac{(-\log(1-p))^k}{k!} \right] + p \sum_{k=0}^{d-1} \left[\frac{(-\log(p))^k}{k!} \right] - 1.$$

Remark 5.3 *The function $h_d(p) = \mu(\mathbb{W}(p))$ is increasing for $p \in [0, 1/2]$ and decreasing for $p \in [1/2, 1]$. Indeed,*

$$h'_d(p) = \frac{1}{(d-1)!} \left[1 - \left(\frac{\log(1-p)}{\log(p)} \right)^{d-1} \right].$$

This means that the construction of $\mathbb{W}(p)$ is very informative for p close to 0 or 1 and the quantity of information is minimal for $p = 1/2$.

It must be noticed that the choice of $(\mathbf{x}^-, \mathbf{x}^+) \in \mathbb{W}^-(p) \times \mathbb{W}^+(p)$ is independent of $\mu(\mathbb{W}(p))$. Then, without more information on g, Equations (5.7) and (5.8) provides only an arbitrary choice for $\mathbf{x}^-$ and $\mathbf{x}^+$.

Corollary 5.1 *Let $p \in]0,1[$ and let q be the p-quantile of $g(\mathbf{X})$, it comes*

$$\mathbb{W}^-(p) \subset \mathbb{U}^- \subset [0,1]^d \backslash \mathbb{W}^+(p).$$

Corollary 5.2 *At most one point of Γ is in $\mathbb{W}^+(p)$.*

Remark 5.4 *The measure of $\mathbb{W}(p)$ tends to 1 as the dimension goes to the infinity. For all $p \in]0,1[$, $\mu(\mathbb{W}(p))$ increases as d increases and converges toward 1 while d goes to infinity.*

In Figure 5.3, the set $\mathbb{W}(p)$ is represented for different values of p in dimension 2. Recall, this set is built without any call to the numerical code g and depends only on p. Indeed, the delimitation of $\mathbb{W}(p)$ is based only on the volume constraint given by Proposition 5.3.

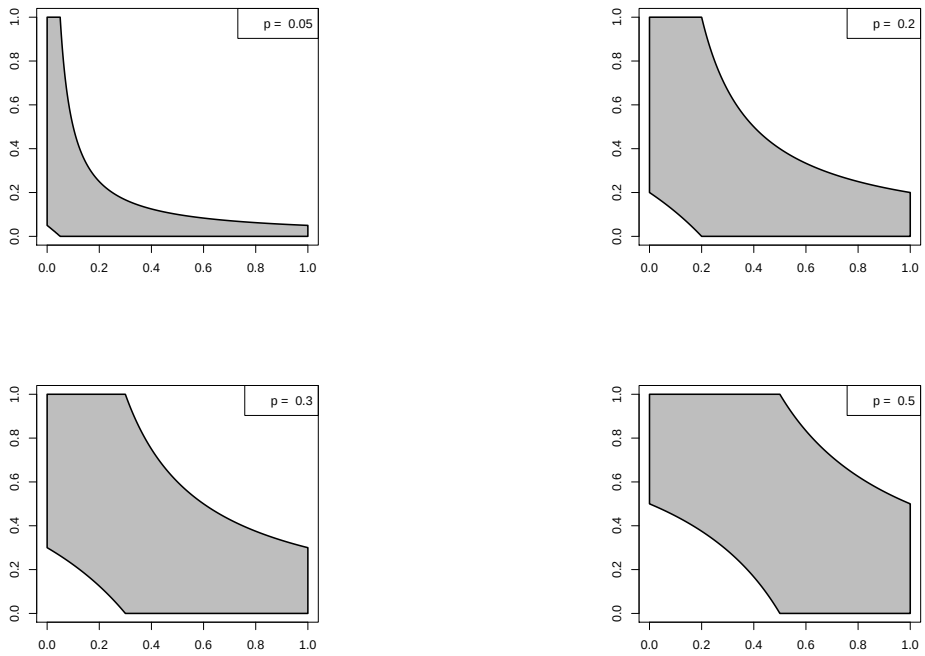


Figure 5.3: The set $\mathbb{W}(p)$ is represented in gray for different values of p in dimension 2.

5.3.2 Updating deterministic bounds

In the previous section, the non-dominated set $\mathbb{W}(p)$ is delimited and two bounds can be obtained by the evaluation of g on two points belonging to the boundary of $\mathbb{W}(p)$. This initialisation is based on the construction of a monotonic set built from a single design point. Following Proposition 5.3, a method to update these bounds is provided in this section. Consider a set of points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $[0,1]^d$. As for the initialisation step, the boundary of $\mathbb{V}^-(\bar{\mathbf{x}}_n)$ (*resp.* $\mathbb{V}^+(\bar{\mathbf{x}}_n)$) is a monotonic set, and then

$$\max_{\mathbf{x} \in \mathbb{V}^-(\bar{\mathbf{x}}_n)} g(\mathbf{x}) \in \{g(\mathbf{x}_1), \ldots, g(\mathbf{x}_n)\},$$
$$\min_{\mathbf{x} \in \mathbb{V}^+(\bar{\mathbf{x}}_n)} g(\mathbf{x}) \in \{g(\mathbf{x}_1), \ldots, g(\mathbf{x}_n)\}.$$

The following proposition establishes a condition on $\bar{\mathbf{x}}_n$ to get respectively an upper and lower bounds for q.

Proposition 5.5 *Let* $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ *in* $[0,1]^d$.

(i) *If* $\mu(\mathbb{V}^-(\mathbf{x}_1, \ldots, \mathbf{x}_n)) \geq p$, *there exists at least one* $m \in [1,n]$ *and* $i_1, \ldots, i_m \in [1,n]$ *such that* $q \leq \max_{\mathbf{x} \in \{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m}\}} g(\mathbf{x})$.

(ii) *If* $\mu(\mathbb{V}^+(\mathbf{x}_1, \ldots, \mathbf{x}_n)) \geq 1 - p$, *there exists at least one* $m \in [1,n]$ *and* $i_1, \ldots, i_m \in [1,n]$ *such that* $q \geq \min_{\mathbf{x} \in \{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m}\}} g(\mathbf{x})$.

Unfortunately, Proposition 5.5 is not constructive since it does not provide neither the value of m nor the set $\{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_m}\}$. Moreover, if the two volume constraints are verified, Proposition

5.5 states that q can be bounded by the minimum and the maximum of g on this numerical design.

The main result of this section is now presented. It consists in choosing a subset $\bar{\mathbf{u}}$ of the numerical design $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ such that $\mu(\mathbb{V}^-(\bar{\mathbf{u}}))$ (*resp.* $\mu(\mathbb{V}^+(\bar{\mathbf{u}}))$) is greater and as close as possible to p (*resp.* $1-p$). To do this, each points of $\bar{\mathbf{u}}$ are chosen sequentially as follows: a point is taken from the numerical design such that the contribution of the volume $\mu(\mathbb{V}^-(\bar{\mathbf{u}}))$ (*resp.* $\mu(\mathbb{V}^+(\bar{\mathbf{u}}))$) for the lower (*resp.* upper) bound is minimum. This procedure is detailed in Algorithms 5.1 and 5.2, and is illustrated on Figure 5.4.

Algorithm 5.1 Getting a lower bound for q from a set of points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

1. Initialisation step : choose $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\dots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^d (1-x^i)$
   Set $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ and $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
2. Choose the next point as $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^+(\bar{\mathbf{x}}_n))$
3. Set $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ and $vol = \mu(\mathbb{V}^+(\bar{\mathbf{u}}))$
4. If $vol \geq 1-p$, repeat steps 2 and 3.
5. Return $\min_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.

Algorithm 5.2 Getting a greater bound for q from a set of points $\bar{\mathbf{x}}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$

1. Initialisation step : choose $\bar{\mathbf{u}} = \underset{\mathbf{x}=(x^1,\dots,x^d)\in\bar{\mathbf{x}}_n}{\arg\min} \prod_{i=1}^d x^i$
   Set $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \bar{\mathbf{u}}$ and $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$
2. Choose the next point as $\mathbf{u} = \underset{\mathbf{x}\in\bar{\mathbf{x}}_n}{\arg\min}\ \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n \cup \mathbf{x})) - \mu(\mathbb{V}^-(\bar{\mathbf{x}}_n))$
3. Set $\bar{\mathbf{u}} = \bar{\mathbf{u}} \cup \mathbf{u}$, $\bar{\mathbf{x}}_n = \bar{\mathbf{x}}_n \backslash \mathbf{u}$ and $vol = \mu(\mathbb{V}^-(\bar{\mathbf{u}}))$
4. if $vol \geq p$, repeat steps 2 and 3 .
5. Return $\max_{\mathbf{u}\in\bar{\mathbf{u}}} g(\mathbf{u})$.

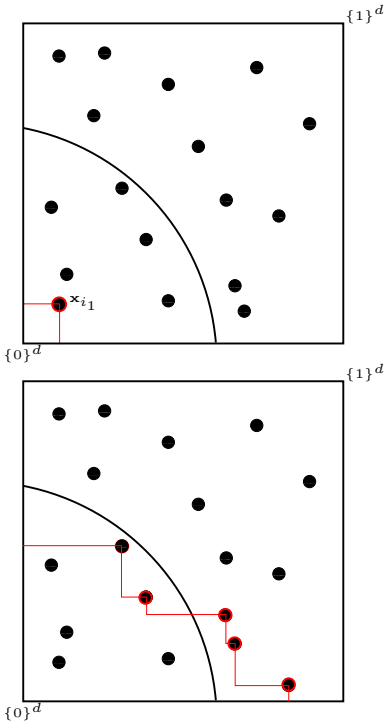
5.4 Sequential quantile estimation

In this section, a consistent estimator of q is provided. The framework follows the one proposed in [16] (see also Section 2.3.1): an initialisation step provides a non-dominated set. This non-dominated set is updated from simulations. This allows to build $\widehat{F}$ a local estimator of F. Finally, q is estimated as in (5.2).

Notice that for probability estimation, only one run to g is required to update the bounds. Indeed, it is enough to know that if g is greater or lower than a given value q. For quantile estimation the situation is more complicated. As seen in Section 5.3.2, the sample must verify one of the two volume constraints stated in Proposition 5.5, which are summarised in Definition 5.3.

Definition 5.3 *Let $A \subset [0,1]^d$. The set A is said p-dominant if one of the two followings properties is verified:*

$$\mu(\mathbb{V}^-(A)) \geq p,$$
$$\mu(\mathbb{V}^+(A)) \geq 1-p.$$



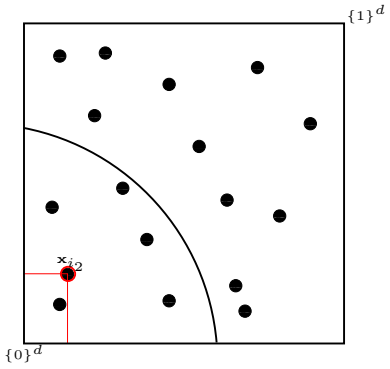


Figure 5.4: Illustration of Algorithm 5.2 for $d = 2$. **Up**: the encircled point $\mathbf{x}_{i_1}$ minimises the contribution of the volume. **Down**: the algorithm is stopped when $\mu(\mathbb{V}^-(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k})) \geq p$. It is required to compute the value of g only on the encircled points.

Assume that the initialisation step described in 5.3.1 is done. Set $(\mathbf{x}^-, \mathbf{x}^+)$ respectively in $\mathbb{W}^-(p)$ and $\mathbb{W}^+(p)$. For example, choose $\mathbf{x}^-$ and $\mathbf{x}^+$ as in (5.7) and (5.8). From these two points, the initial bounds for q are defined as follow

$$g(\mathbf{x}^-) = q_0^- \leq q \leq q_0^+ = g(\mathbf{x}^+).$$

Moreover, let $\mathbb{U}_0 = \mathbb{W}(p), \mathbb{U}_0^- = \mathbb{W}^-(p)$ and $\mathbb{U}_0^+ = \mathbb{W}^+(p)$ as defined in Proposition 5.4 and denote p_0^-, p_0^+ respectively the Lebesgue measures of $\mathbb{U}_0^-$ and $[0,1]^d \backslash \mathbb{U}_0^+$.

The framework is detailed for the first step then at a general step. Let $\mathbf{X}_1$ be uniformly distributed on $\mathbb{U}_0$. Let us start this step by updating the non-dominated set. If $g(\mathbf{X}_1) \leq q_0^-$ (resp. $g(\mathbf{X}_1) \geq q_0^+$), from the monotonicity of g it is deduced that $\mathbf{X}_1 \in \mathbb{U}^-$ (resp. $\mathbf{X}_1 \in \mathbb{U}^+$). The non-dominated set is then updated as follow

$$\mathbb{U}_1^- = \begin{cases} \mathbb{U}_0^- \cup \mathbb{V}^-(\mathbf{X}_1) & \text{if } g(\mathbf{X}_1) \leq q_0^-, \\ \mathbb{U}_0^- & \text{if } g(\mathbf{X}_1) > q_0^-, \end{cases}$$

$$\mathbb{U}_1^+ = \begin{cases} \mathbb{U}_0^+ \cup \mathbb{V}^+(\mathbf{X}_1) & \text{if } g(\mathbf{X}_1) \geq q_0^+, \\ \mathbb{U}_0^+ & \text{if } g(\mathbf{X}_1) < q_0^+, \end{cases}$$

$$\mathbb{U}_1 = [0,1]^d \backslash \left(\mathbb{U}_1^- \cup \mathbb{U}_1^+ \right),$$

and $p_1^- = \mu(\mathbb{U}_1^-), p_1^+ = 1 - \mu(\mathbb{U}_1^+)$.

Unfortunately, the bounds of the quantile cannot be updated at this first step. Indeed, at least two points in the non-dominated set are required to verify the p-dominant property. Since the point $\mathbf{X}_1$ cannot be used to update the bounds of q, it is kept for the following steps. Let $\bar{\mathbf{X}}_1 = \{\mathbf{X}_1\} \cap \mathbb{U}_1$ the remaining points in the non-dominated set. Thus, $q_1^- = q_0^-$ and $q_1^+ = q_0^+$. Concluding on this first step, an estimator of q is now described. Denote

$$\widehat{F}_1(t) = p_0^- + (p_0^+ - p_0^-) \mathbb{1}_{\{g(\mathbf{X}_1) \leq t\}}.$$

Using the same argument as in Section 4.2, it comes $\mathbb{E}\left[\widehat{F}_1(q)\right] = F(q)$ and q is estimated as follow

$$\widehat{q}_1 = \inf\{t \in [q_1^-, q_1^+],\ \widehat{F}_1(t) \geq p\}.$$

At step $n \geq 1$, denote

$$\mathcal{F}_{n-1} = \sigma(\mathbf{X}_j, 0 \leq j \leq n-1),$$

and let $\mathbf{X}_n$ be a random vector uniformly distributed on $\mathbb{U}_{n-1}$. If $g(\mathbf{X}_n)$ is in $]q_{n-1}^-, q_{n-1}^+[$ then the non-dominated set is updated as described in step 1. Otherwise, it must be checked if the set $\bar{\mathbf{X}}_{n-1} \cup \{\mathbf{X}_n\}$ is p-dominant. If so, Algorithms 5.1 and 5.2 are applied and the bounds q_{n-1}^-, q_{n-1}^+ are updated as well as the non-dominated set. Denote, $\bar{\mathbf{X}}_n = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\} \cap \mathbb{U}_n$. An unbiased estimator of $F(q)$ becomes

$$\widehat{F}_n(q) = \frac{1}{n} \sum_{k=1}^{n} \left(p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-) \mathbb{1}_{\{g(\mathbf{X}_k) \leq q\}} \right), \tag{5.10}$$

and finally, q is estimated by

$$\widehat{q}_n = \inf \left\{ t \in [q_n^-, q_n^+],\ \widehat{F}_n(t) \geq p \right\}. \tag{5.11}$$

The two following propositions provide some properties of the estimator of the cumulative distribution function and ensure that the bounds and the estimator of the quantile converge towards q.

Proposition 5.6 *Let $(q_n^-, q_n^+)_{n \geq 1}$ be the sequence of bounds obtained from the method. Almost surely*

$$(q_n^-, q_n^+) \underset{n \to +\infty}{\longrightarrow} (q, q),$$

and consequently $\widehat{q}_n \underset{n \to +\infty}{\overset{a.s.}{\longrightarrow}} q$.

Proposition 5.7 *For all $t \in \mathbb{R}$, $F(\min(t, q)) \leq \mathbb{E}\left[\widehat{F}_n(t)\right] \leq F(\max(t, q))$ and*

$$\mathbb{E}\left[\widehat{F}_n(t)\right] \underset{n \to +\infty}{\longrightarrow} F(q).$$

The main difference with probability estimation framework, is that the Lebesgue measure of the non-dominated set is not strictly decreasing. Indeed, it cannot be updated after each evaluation by g since either the set of simulations must be p-dominant or the value of g on these simulations is between the bounds of q.

As a central limit theorem on the probability estimator is not available, showing a central limit theorem for $\widehat{q}_n$ seems difficult. Another important point is the rate of convergence of the deterministic bounds of q. Nonetheless, this rate depends on Algorithms 5.1 and 5.2, it is not obvious therefore difficult to get precise results in a general case.

5.5 Numerical results

In this section, the method described in this chapter is compared with other classical quantile estimators. The comparison is conducted on an analytic example where the dimension and the probability can be chosen. Let $d \geq 2$ and $\mathbf{x} = (x^1, \ldots, x^d)$ in $[0,1]^d$, the function is defined by

$$g(\mathbf{x}) = \frac{x^1}{\sum_{i=1}^{d} x^i}.$$

Let $\mathbf{X} = (X^1, \ldots, X^d)$ be a random vector with independent coordinates such that $X^i \sim \Gamma(i+1, 1)$. Then $g(\mathbf{X}) \sim \mathcal{B}eta(2, (d+1)(d+2)/2 - 3)$. Denote $q_{d,p}$ the p-quantile of $g(\mathbf{X})$. The quantile estimator is compared on six different couples $(d,p) \in \{2,5,7\} \times \{10^{-2}, 10^{-4}\}$.

The comparison is conducted with two different estimators based on an iid sample and the sequential framework. The first two criteria are the relative error defined by $\mathbb{E}[|\alpha - q_{d,p}|/q_{d,p}]$ and the mean quadratic error defined by $\mathbb{E}[(\alpha - q_{d,p})^2]$ where α is one of the estimators. The last two criteria are the bias and the length of the empirical confidence interval at 95%. Even if the function is analytic, consider that the total number of evaluations by g is very limited. Arbitrary, a budget of $n = 200$ simulations have been chosen.

First, the method is tested for $p = 10^{-2}$. Figure 5.5 displays the results obtained for $p = 10^{-2}$ in function of $n = 1, \ldots, 200$ and averaged on 800 independent experiments. From left to right these results are obtained respectively for $d = 2, 5, 7$. The sequential estimator is better, but its performance seems to decrease while the dimension increases. The sequential quantile estimator depends on an estimation of the cumulative distribution function of $g(\mathbf{X})$ which depends itself on the distances between each bounds p_{k-1}^-, p_{k-1}^+ and the probability p. As discussed in the previous chapter, this distance between the bounds increases with the dimension.

The rate of convergence of the quantile bounds depends also on the value of g. Indeed, if $g(\mathbf{X})$ is uniformly distributed on $[0,1]$, the two bounds obtained by the initialisation, described in Section 5.3.1, are both equal to q. The sequential estimator is still efficient for $d = 7$. Let us look now how the sequential estimator behave when p is smaller.

Table 5.1 provides the same quantities as Figure 5.5 for $n = 200$ and $p = 10^{-4}$, here the results have been averaged on 137 independent experiments. The sequential estimator is still efficient for a low probability. This can be explained by Remark 5.3: the initialisation step is more informative when p becomes far from $1/2$. The quadratic error of the two estimators decreases when the dimension increases. This can be explained by the difficulty to estimate the cumulative distribution function of $g(\mathbf{X})$. Indeed, this becomes more and more difficult when the dimension increases. The constructions of these estimators are very different and this reduction can be explained by the value taken by the numerical code around the limit state. Finally, the accuracy of the estimator based on a sequential framework seems consistent when the probability becomes small.

In addition to compare the estimators, Table 5.2 provides the value of $\mathbb{E}[(q_n - q)^2]$ where q_n is the lower or the upper bound obtained by standard Monte Carlo or the sequential strategy for the different couples (d,p). When the dimension increases, the gain obtained from a sequential strategy strongly diminishes for $p = 10^{-4}$.

To conclude on this example, for each couple (d,p) the sequential strategy provides more information than standard Monte Carlo: the estimator is more accurate and the exact bounds converge faster. Nonetheless, the construction of this estimator requires the computation of the volume of the dominated set. This computation becomes more and more difficult when the dimension increases. When the budget of simulations is very limited and when the q is a very

	Methods	$d=2$	$d=5$	$d=7$
Relative error	Seq	0.15	0.60	1.67
	MC	5.53	5.26	5.70
Quadratic error	Seq	2.00×10^{-6}	3.62×10^{-7}	8.18×10^{-7}
	MC	7.39×10^{-4}	2.41×10^{-5}	8.14×10^{-6}
Bias	Seq	3.71×10^{-4}	2.96×10^{-4}	6.76×10^{-4}
	MC	2.26×10^{-2}	4.02×10^{-3}	2.41×10^{-3}
Length of the empirical confidence interval at 95%	Seq	3.99×10^{-3}	1.85×10^{-3}	1.91×10^{-3}
	MC	5.18×10^{-2}	1.13×10^{-2}	1.12×10^{-2}

Table 5.1: Toy example for $p = 10^{-4}$ at step $n = 200$. Comparison of the relative error, the quadratic error, the bias and the length of the empirical confidence interval at 95% of different estimators. Results have been averaged on 137 independent experiments.

		Methods	$d=2$	$d=5$	$d=7$
$p = 10^{-2}$	Lower bound	Seq	7.41×10^{-5}	4.56×10^{-5}	1.46×10^{-5}
		MC	$-$	$-$	$-$
	Upper bound	Seq	5.75×10^{-5}	1.82×10^{-4}	2.39×10^{-4}
		MC	1.29×10^{-3}	6.72×10^{-4}	5.28×10^{-4}
$p = 10^{-4}$	Lower bound	Seq	6.64×10^{-6}	4.67×10^{-7}	1.46×10^{-7}
		MC-low	$-$	$-$	$-$
	Upper bound	Seq-up	8.80×10^{-7}	8.69×10^{-6}	1.80×10^{-5}
		MC-up	9.17×10^{-4}	6.44×10^{-5}	3.82×10^{-5}

Table 5.2: Toy example for $p = 10^{-4}$ and at step $n = 200$. Quadratic error defined by $\mathbb{E}[(q_n - q)^2]$ with q_n the lower or the upper bound obtained by a Monte Carlo sample or by a sequential framework. The symbol "$-$" means that the bound is not available with the sample obtained.

low quantile, a sequential framework appears clearly more adapted than an independent and identically distributed sample.

5.6 Conclusion

This chapter provides a first method to bound exactly a quantile under monotonic assumption. Using this bounding method a consistent estimator of the quantile has been obtained from a sequential Monte Carlo sampling. First, numerical results show that the sequential framework decreases significantly the bias and the quadratic error of the estimator and accelerate the convergence of the quantile bounds. Since the quantile estimator depends on the bounds of p, the accuracy of the estimator decreases while the dimension increases. When the probability goes to 0, the initialisation step reduces the volume of the non-dominated set. The sequential estimator is more adapted for low probability.

The main limit of this work is that only one quantile may be estimated at a time. The simulations used to estimate one quantile can be exploited as an initialisation step to bounds any other quantile. Controlling the estimator by a central limit theorem seems difficult since it has not been obtained for probability estimation. In this chapter, a p-monotonic set has been obtained from a set of points. Building a more regular surface is not usable to get an upper

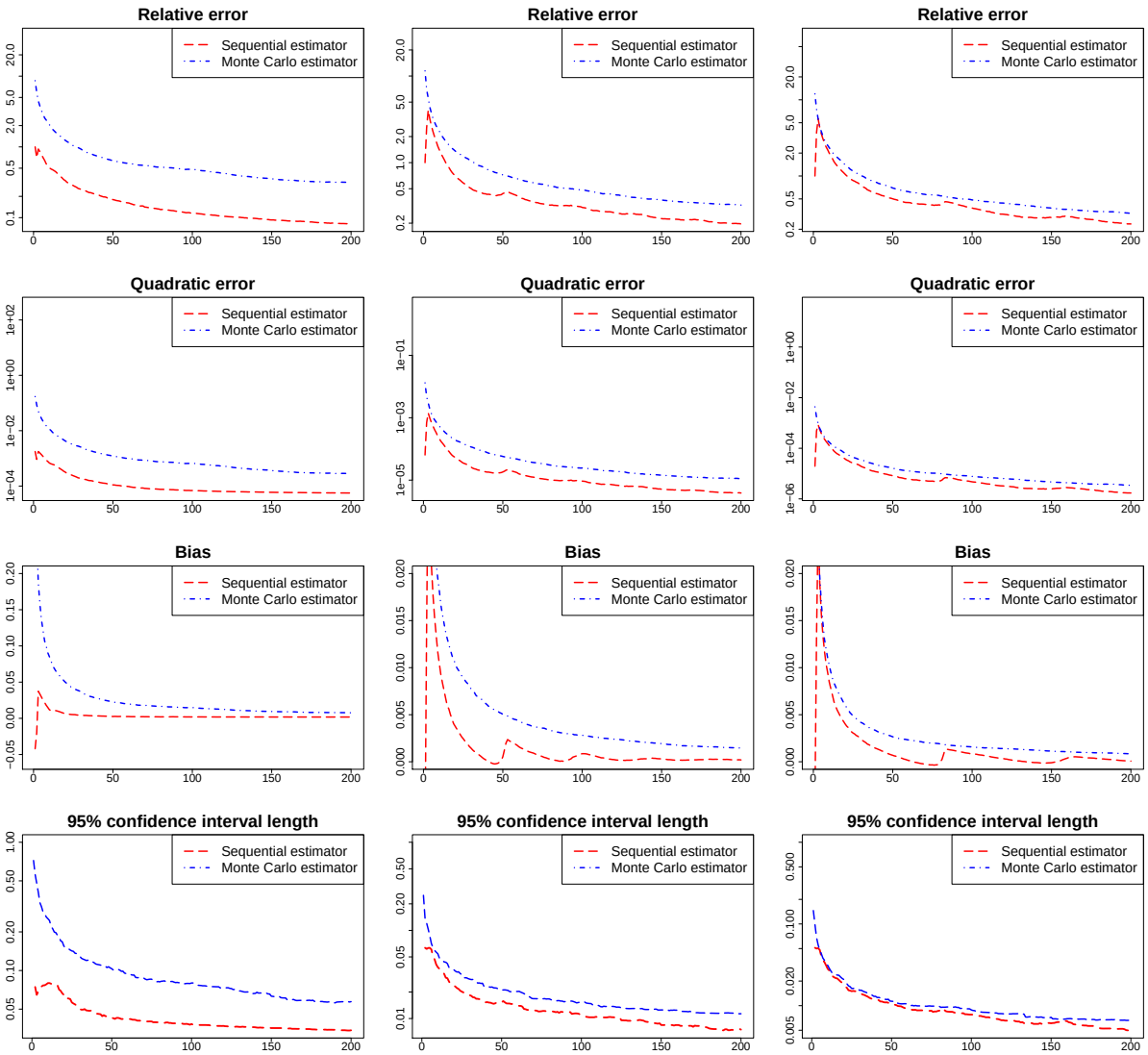


Figure 5.5: Toy example for $p = 10^{-2}$, from left to right $d = 2, 5, 7$. Comparison of the relative error, the quadratic error, the bias and the length of the empirical 95% confidence interval of different estimators in function of $n = 1, \ldots, 200$. Results have been averaged on 800 independent experiments.

bound for q since a finite number of g is possible. Nonetheless, the use of a meta-model to estimate Γ allows to guide the construction of a design of experiment. Controlling the rate of convergence of the quantile for different framework can be useful to control the estimator.

5.7 Proofs

Proof of Proposition 5.1. Let S_p be p-monotonic set. If $S_p = \Gamma_q$ the proof is straightforward. Assume that $S_p \neq \Gamma_q$. Since $\mathbb{V}^-(S_p) = p$, the set S_p is not strictly contained in $\mathbb{U}_q^-$. Then, there exist $\mathbf{x}, \mathbf{y}$ in S_p such that $(\mathbf{x}, \mathbf{y}) \in \mathbb{U}_q^- \times \mathbb{U}_q^+$. Thus, $g(\mathbf{x}) \leq q \leq g(\mathbf{y})$. The proof is complete choosing the minimum and the maximum of g on S_p. □

Proof of Proposition 5.2. Let $f \in C$ such that for all $\mathbf{x} \in [0,1]^{d-1}$, $f(\mathbf{x}) \in]0,1[$. The function$f$ is not an extremal element of C since there exist $\varepsilon > 0$ such that for all $\mathbf{x} \in [0,1]^{d-1}$, $f(\mathbf{x}) = \frac{f(\mathbf{x})+\varepsilon}{2} + \frac{f(\mathbf{x})-\varepsilon}{2}$.

Let $f \in C$ and such that $f \in]0,1]$. Denote $\lambda = f(\mathbf{0}) \in]0,1]$ and let $g, h \in C$ such that

$$g(\mathbf{x}) = \begin{cases} f(\mathbf{x})/\lambda & \text{if } f(\mathbf{x}) > 0, \\ 0 & \text{otherwise} \end{cases},$$
$$h(\mathbf{x}) = 0.$$

The function f is not an extremal element of C since $f = \lambda g + (1-\lambda)h$. Symmetrically, if $f \in C$ and such that $f \in [0,1[$ then f is not an extremal element of C.

Let $f \in C$ such that for all $\mathbf{x} \in [0,1]^{d-1}$, $f(\mathbf{x}) \in \{0,1\}$. Then f is an indicator function on a monotonic set S. There exists a sequence $(P_i)_{i\geq 1}$ of elements of $\mathcal{P}$ such that $S = \cup_{i\geq 1} P_i$. It comes $f = \mathbb{1}_{\cup_{i\geq 1} P_i}$. Let $g, h \in C$ such that $g = \mathbb{1}_A$ and $h = \mathbb{1}_B$.

Assume there exist $\lambda \in]0,1[$ such that $f = \lambda g + (1-\lambda)h$. Let $\mathbf{x} \in \cup_{i\geq 1} P_i$ then $1 = \lambda g(\mathbf{x}) + (1-\lambda) g(\mathbf{x})$. Since $g, h \in \{0,1\}, g(\mathbf{x}) = h(\mathbf{x}) = 1$ and then $\mathbf{x} \in A \cap B$. If $\mathbf{x} \notin \cup_{i\geq 1} P_i$ then $0 = \lambda g(\mathbf{x}) + (1-\lambda) g(\mathbf{x})$ and then $\mathbf{x} \notin A \cup B$. Finally, $A = B = \cup_{i\geq} P_i$ and $f = g = h$.

Since any monotonic can be expressed an union of elements of $\mathcal{P}$ the proof is complete. □

Proof of Proposition 5.4. Let $p \in]0,1[$ and $\mathbf{U} = (U^1, \ldots, U^d) \sim \mathcal{U}([0,1]^d)$, then $\mu(\mathbb{W}^+(p)) = \mathbb{P}(\mathbf{U} \in \mathbb{W}^+(p)) = \mathbb{P}(U^1 U^2 \cdots U^d > p)$. The probability density function of $U^1 U^2 \cdots U^d$ is defined for all $u \in [0,1]$ by

$$f_d(u) = \frac{(-\log u)^{d-1}}{(d-1)!} \mathbb{1}_{\{u \in [0,1]\}}.$$

Let $t \in]0,1[$ and denote $I_d(t) = \mu(\mathbb{W}^+(t)) = \int_t^1 f_d(u) du$. By the substitution $v = -\log u$ and an integration by parts, it comes

$$\begin{aligned} I_d(t) &= \int_t^1 \frac{(-\log u)^{d-1}}{(d-1)!} du, \\ &= \int_0^{-\log t} \frac{v^{d-1} e^{-v}}{(d-1)!} dv, \\ &= I_{d-1}(t) - \frac{(-\log t)^{d-1} t}{(d-1)!}. \end{aligned}$$

By recursion, it comes

$$I_d(t) = 1 - t\sum_{k=0}^{d-1} \frac{(-\log t)^k}{k!}.$$

Since $\mu(\mathbb{W}(p)) = 1 - I_d(p) - I_d(1-p)$ the proof is completed. □

Proof of Proposition 5.6. The proof uses the same argument than the proof of Proposition 3.2. Let $(\mathbf{Y}_k)_{k\geq 1}$ be a sequence of independent random vectors in $[0,1]^d$ such that there exists $\varepsilon_1 > 0$ such that for all $\mathbf{x} \in \Gamma^{\varepsilon_1} = \{\mathbf{x} \in [0,1]^d,\ d(\mathbf{u},\Gamma) < \varepsilon_1\}$ there exists $\varepsilon_2 > 0$ such that

$$\sum_{n\geq 1} \mathbb{P}(\mathbf{Y}_n \in B(\mathbf{x},\varepsilon_2)) = +\infty. \tag{5.12}$$

(1). For all $n \geq 1$, let q_n^- and q_n^+ be the bounds of q obtained from Algorithms 5.1 and 5.2 applied to $\mathbf{Y}_1,\ldots,\mathbf{Y}_n$. The sequences $(q_n^-)_{n\geq 1}$ and $(q_n^+)_{n\geq 1}$ are respectively increasing and decreasing and for all $n \geq 1$, $q_n^- \leq q \leq q_n^+$ almost surely. From construction, there exists two random variables q_∞^-, q_∞^+ such that almost surely

$$\begin{aligned} q_n^- &\underset{n\to+\infty}{\longrightarrow} q_\infty^- \leq q, \\ q_n^+ &\underset{n\to+\infty}{\longrightarrow} q_\infty^+ \geq q. \end{aligned}$$

For all $n \geq 1$, $q_n^-, q_n^+ \in \{g(\mathbf{Y}_1),\ldots,g(\mathbf{Y}_n)\}\}$ and then $\mathbf{Y}_k \notin \Gamma = \{\mathbf{x} \in [0,1]^d, g(\mathbf{x}) = q\}$ for all k. Denote for all $n \geq 1$

$$\begin{aligned} \mathbb{U}_n^- &= \bigcup_{\mathbf{x}\in\{\mathbf{Y}_1,\ldots,\mathbf{Y}_n\},\ g(\mathbf{x})\leq q_n^-} \mathbb{U}^-(\mathbf{x}), \\ \mathbb{U}_n^+ &= \bigcup_{\mathbf{x}\in\{\mathbf{Y}_1,\ldots,\mathbf{Y}_n\},\ g(\mathbf{x})\geq q_n^+} \mathbb{U}^-(\mathbf{x}). \end{aligned}$$

Assume that $q_\infty^- < q < q_\infty^+$, then

$$\begin{aligned} \mathbb{U}_\infty^- &\bigcup_{\mathbf{x}\in\{\mathbf{Y}_1,\ldots,\mathbf{Y}_n,\ldots\},\ g(\mathbf{x})\leq q_\infty^-} \mathbb{U}^-(\mathbf{x}) \subsetneq \mathbb{U}^-, \\ \mathbb{U}_\infty^+ &\bigcup_{\mathbf{x}\in\{\mathbf{Y}_1,\ldots,\mathbf{Y}_n,\ldots\},\ g(\mathbf{x})\geq q_\infty^+} \mathbb{U}^-(\mathbf{x}) \subsetneq \mathbb{U}^+, \end{aligned}$$

and denote $\mathbb{U}_\infty = [0,1]^d \backslash (\mathbb{U}_\infty^- \cup \mathbb{U}_\infty^+) \supset \Gamma$. From assumption on the sequence $(\mathbf{Y}_k)_{k\geq 1}$ and from Borel-Cantelli Lemma, the event $A_k = \{\mathbf{Y}_k \in \mathbb{U}_\infty\}$ occurs infinitely often. Denote $(\mathbf{Y}_k^{(\infty)})_{k\geq 1}$ the subsequence of $(\mathbf{Y}_k)_{k\geq 1}$ which are in $\mathbb{U}_\infty$ and

$$M_\infty = \inf\{k \geq 1,\ \mu(\mathbb{U}^-(\mathbf{Y}_1^{(\infty)},\ldots,\mathbf{Y}_k^{(\infty)})) \geq p\}.$$

Since $\{M_\infty = +\infty\} = \{\forall k \geq 1, \mathbf{Y}_k^{(\infty)} \in \mathbb{U}^- \backslash \Gamma\}$, the Borel-Cantelli lemma applied to $B_k = \{\mathbf{Y}_k^{(\infty)} \in \mathbb{U}_\infty \cap \mathbb{U}^+\}$ states that $\mathbb{P}(M_\infty = +\infty) = 0$. This means the sequence $(\mathbf{Y}_k^{(\infty)})_{k\geq 1}$ is almost surely p-dominant and then the upper bound q_∞^+ can be updated and then $q_\infty^+ = p$. Using the same argument for the sequence q_∞^- the sequence of bounds $(q_n^-, q_n^+) \underset{n\to+\infty}{\longrightarrow} q$ almost surely.

(2). For the sequential case, denote $T_0 = 0$ and for all $n \geq 1$, $T_n = \inf\{k \geq T_{n-1}, \mathbf{Y}_k \in \mathbb{U}_{T_{n-1}}\}$. The bound obtained in a sequential framework are given by $\tilde{q}_n^- = q_{T_n}^-, \tilde{q}_n^+ = q_{T_n}^+$ and converges towards q . From construction, $\tilde{q}_n^- \leq \widehat{q}_n \leq \tilde{q}_n^+$ then $\widehat{q}_n \underset{n\to+\infty}{\longrightarrow} q$ almost surely.

The proposition is proven if for all $k \geq 1, \mathbf{Y}_k$ uniformly distributed on $[0,1]^d$. Condition (5.12) is verified and apply **(1)** and **(2)** to the sequence $(\mathbf{Y}_k)_{k\geq 1}$ conclude the proof. □

Proof of Proposition 5.7. Let us start with $t > q$.

$$\begin{aligned}
\mathbb{E}\left[\widehat{F}_n(t)\right] &= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{g(\mathbf{X}_k)\leq t\}}\right], \\
&= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{E}[\mathbb{1}_{\{g(\mathbf{X}_k)\leq t\}}|\mathcal{F}_{k-1}]\right], \\
&= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\frac{\mu(\{\mathbf{x}, g(\mathbf{x}) \leq t\} \cap \mathbb{U}_{k-1})}{p_{k-1}^+ - p_{k-1}^-}\right], \\
&= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + \mu(\{\mathbf{x}, g(\mathbf{x}) \leq t\} \cap \mathbb{U}_{k-1})\right].
\end{aligned}$$

Since $t > q$, it comes

$$\begin{aligned}
\mathbb{E}\left[\widehat{F}_n(t)\right] &\leq \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + \mu\left(\{\mathbf{x}, g(\mathbf{x}) \leq t\} \cap \left(\mathbb{U}_{k-1} \cup \mathbb{U}_{k-1}^+\right)\right)\right], \\
&= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + F(t) - p_{k-1}^-\right], \\
&= F(t).
\end{aligned}$$

Moreover,

$$\begin{aligned}
\mathbb{E}\left[\widehat{F}_n(t)\right] &\geq \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + \mu\left(\{\mathbf{x}, g(\mathbf{x}) \leq q\} \cap \mathbb{U}_{k-1}\right)\right], \\
&= \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + F(q) - p_{k-1}^-\right], \\
&= F(q).
\end{aligned}$$

Since

$$\begin{aligned}
\mathbb{E}\left[\mu(\{\mathbf{x}, g(\mathbf{x}) \leq t\} \cap \mathbb{U}_{k-1})\right] &\leq \mathbb{E}\left[\mu(\mathbb{U}_{k-1})\right], \\
&= \mathbb{E}[p_{k-1}^+ - p_{k-1}^-], \\
&\underset{n\to+\infty}{\longrightarrow} 0,
\end{aligned}$$

and $\mathbb{E}\left[p_{k-1}^-\right] \underset{k\to+\infty}{\longrightarrow} p$ from Proposition 5.12. Toeplitz lemma proves that $\mathbb{E}\left[\widehat{F}_n(t)\right] \underset{n\to+\infty}{\longrightarrow} F(q)$. Using the same arguments for $t < q$ and that $\mathbb{E}\left[\widehat{F}_n(q)\right] = F(q)$ finish the proof. □

Chapter 6

Industrial case study

In this section, the industrial numerical that has motivated this work is described. It is considered a non-replaceable component of a pressurized water vessel. Such component is subject to different physical constraints. For example, the irradiation from the nuclear reaction leads to a loss of matter. Of course, this loss of matter has a harmful effect on the reliability of the component. Moreover, the extreme temperature and pressure of its environment could cause some deformation of it. Manufacturing defects must be taking into account.

The numerical model associated to this component represents the reliability according to the propagation of a flaw. In such extreme condition, this flaw may spread in the component. It must be justified that this eventual spreading does not involve the loss of integrity of the component.

Some characteristics are taking into account in this study and are summarized in Table 6.1.

This numerical code is considered black-box, but it is not so time-consuming. Approximately one second is required to make one run. The input of this numerical code depends on many parameters, but most of them are fixed variables. Determine the monotonicity of g has not been studied in this thesis. Nonetheless, a first work has been recently conducted in [9] to provide information on the monotonicity of a function. Finally, the knowledge of the monotonicity has been obtained by expert opinions. It is then considered that the monotonic properties of the numerical code are totally known.

In this study, four inputs are considered as independent random variables. They are denoted $\mathbf{X} = (X^1, X^2, X^3, X^4)$ with a known probability density function given in Table 6.1

In order to compare the reliability of different situations, two cases are studied. In the first one, it is considered that only X^1 and X^2 are represented by a random vectors and X^3, X^4 are fixed to their nominal value. In the second one, $\mathbf{X} = (X^1, X^2, X^3, X^4)$ is represented by a random vector. As the monotonicity of g is known as well as the probability density function of its input, it is transformed to a globally increasing function (see Section 2.2).

In Figure 6.1 the value of the numerical code is represented on a sample distributed as

Input	Distribution	parameters	Physical representation
X^1	truncated Weibull	(1.8, 0.00309 , 0.00005, 0.05)	depth of a flaw
X^2	Log-normal	(-1.516, 0.504)	Ratio height/length
X^3	Normal	(0,1)	Resistance
X^4	Normal	(0,1)	Constraints

Table 6.1: Input of the industrial numerical code.

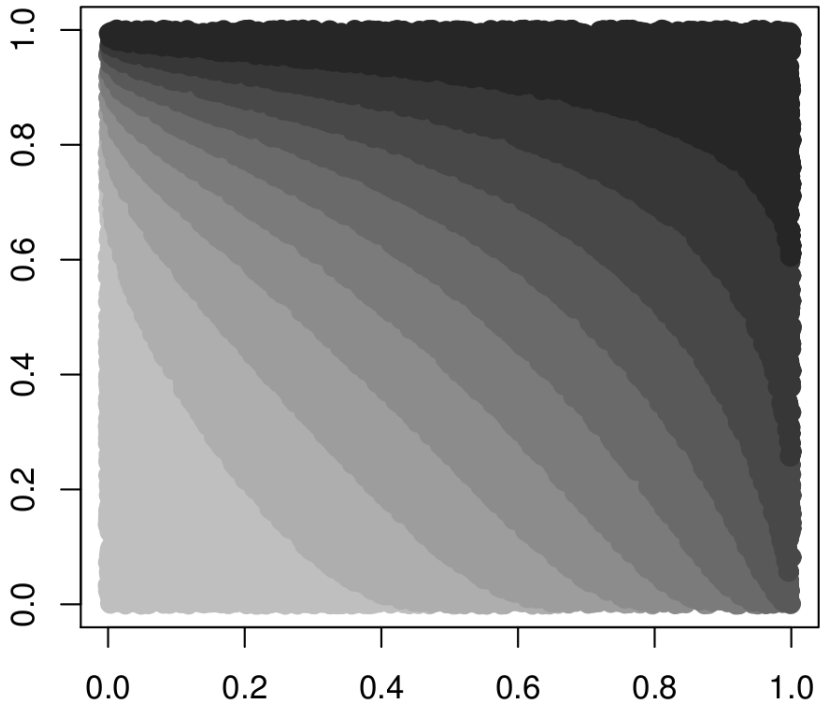


Figure 6.1: Representation of the outputs of the real case study in dimension 2.

(X^1, X^2). It must be noticed that this figure has been obtained after the transformation (2.1). Each set of different colours delimits one of the ten decile of $g(\mathbf{X})$.

As studied in this thesis, two indicators are examined. The first one is the probability that the numerical code is lower than a given threshold q:

$$\mathbb{P}(g(\mathbf{X}) \leq q).$$

The second indicator is a quantile q of the values taken by the numerical code. The value of q increases with the reliability of the component. As for probability estimation, if there is a reference value q_0 for this quantile, it must be proved that q is greater than q_0. Under the monotonic hypothesis, if the deterministic lower bound is greater than q_0 then an estimator of q is no more informative. This ensures that the reliability have been proven.

Probability and quantile estimations are studied in the two following sections.

6.1 Probability estimation

In this section, the MLE and the estimator (4.17) obtained in Chapter 4

$$\widehat{p}_n = \frac{1}{n}\sum_{k=1}^{n} p_{k-1}^{-} + (p_{k-1}^{+} - p_{k-1}^{-})\mathbb{1}_{\mathbf{X}_k \in \mathbb{U}^-},$$

are compared. For each dimensions d considered, two thresholds $q_{d,1} < q_{d,2}$ are given and then two probabilities must be estimated.

6.1.1 Two dimensional case

Figure 6.2 provides results obtained from a sequential uniform distribution strategy. The values of the deterministic bounds, the expectation of estimators and their coefficient of variation defined by

$$CV\left(\widehat{p}_n\right) = \frac{\sqrt{Var(\widehat{p}_n)}}{\mathbb{E}[\widehat{p}_n]},$$
$$CV\left(\widehat{p}_n^{MLE}\right) = \frac{\sqrt{Var(\widehat{p}_n^{MLE})}}{\mathbb{E}[\widehat{p}_n^{MLE}]},$$

are computed in function of n.

For the two thresholds, the MLE and $\widehat{p}_n$ seem to have the same expectation. Since $\widehat{p}_n$ is unbiased, it can be deduced that the MLE has a low bias. In term of variance the MLE outperforms $\widehat{p}_n$ for the two thresholds. This can be explained by the low dimension. Indeed, the bounds seem to be symmetric around the estimate.

6.1.2 Four dimensional case

Consider now the four dimensional case. As for the previous subsection, the bounds obtained as well as the expectation and the coefficient of variation of the two estimators are compared. As before, the two thresholds $q_{4,1}$, $q_{4,2}$ are given and are such that $q_{4,1} < q_{4,2}$. For different values of $n, CV\left(\widehat{p}_n^{MLE}\right)$ is lower than $CV\left(\widehat{p}_n\right)$. The two estimators provide equivalent estimation for $q = q_{4,2}$. Nevertheless, for $q = q_{4,1}$ they are slightly different. In this case, the deterministic bounds are symmetric around the estimate. Then, the bias of the MLE can be explained by the low probability to be estimated and the increasing dimension. As seen in Chapter 4, the MLE is no longer efficient when the dimension increases and p decreases. To conclude, $\widehat{p}_n^{MLE}$ seems to be more appropriate to use in practice when the dimension is low. Its bias seems smaller in such situations. Nonetheless, the performance of $\widehat{p}_n$ seems to be less influenced by the dimension and the magnitude of p.

6.2 Quantile estimation

In this section, quantile estimation proposed in (5.11) is compared with the estimator based on the standard Monte Carlo sampling. For the deterministic bounds, it is compared the gain of a sequential design experiments with an iid sample. The comparison is conducted for $p \in \{10^{-5}, 10^{-3}\}$. A reference value of q_p with $p = 10^{-3}$ is provided by the standard Monte Carlo estimator. Getting such reference value, a sample of size 10^5 has been produced. For $p = 10^{-5}$, too many simulations are required to have an accurate estimation of q_p by a standard Monte Carlo method. This case is studied to see how to performs the estimator with a very small quantile. Moreover, all results have been normalised to be in $[0, 1]$.

6.2.1 Two dimensional case

Estimation of a $p = 10^{-3}$-quantile

In Figure 6.3, the bias, the quadratic error and the variance of the studied estimators are illustrated in function of the number of evaluations n made by the numerical code. The sequential estimator is significantly better than the other ones for different quantities of interest. The bias

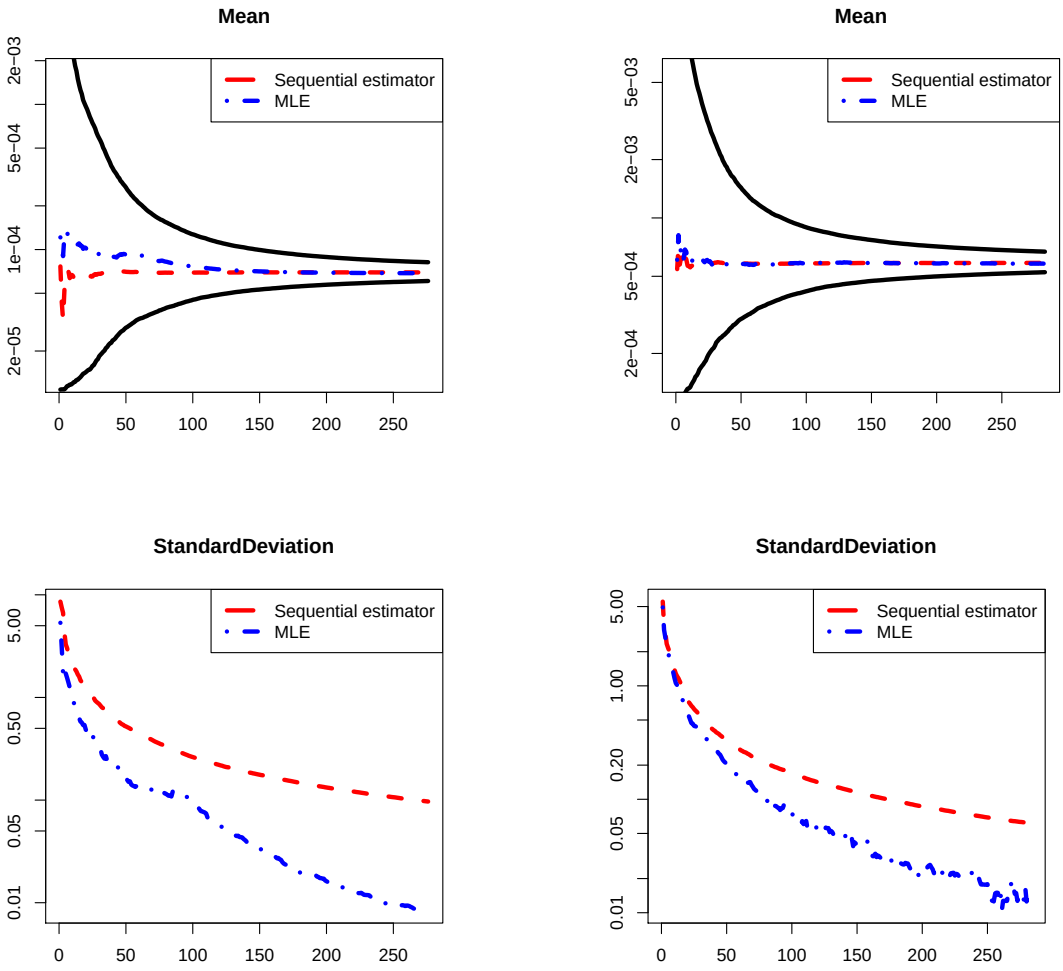


Figure 6.2: Industrial case for $d = 2$. It is compared the mean of each estimators and their coefficient of variation in function of n. In left for $q_{2,1}$ and right for $q_{2,2}$.

	Methods	$n = 100$	$n = 200$	$n = 300$
$q_{4,1}$	$p_n^- \times 10^{-4}$	0.0046	0.0062	0.0087
	$p_n^+ \times 10^{-4}$	442	240	160
	$\widehat{p}_n^{MLE} \times 10^{-4}$	1.23	1.25	1.25
	$\widehat{p}_n \times 10^{-4}$	0.724	0.798	0.799
	$CV\left(\widehat{p}_n^{MLE}\right)$	1.694	0.860	0.603
	$CV\left(\widehat{p}_n\right)$	3.208	1.622	1.163
$q_{4,2}$	$p_n^- \times 10^{-3}$	0.015	0.028	0.045
	$p_n^+ \times 10^{-3}$	8.30	4.83	3.31
	$\widehat{p}_n^{MLE} \times 10^{-3}$	1.10	1.044	1.097
	$\widehat{p}_n \times 10^{-3}$	1.10	1.056	1.087
	$CV\left(\widehat{p}_n^{MLE}\right)$	1.084	0.665	0.414
	$CV\left(\widehat{p}_n\right)$	1.148	0.715	0.492

Table 6.2: Industrial case study for $d = 4$. The deterministic bounds, the two estimates and their coefficient variations are given for different values of n. Up (resp. right) the estimation has been made for $q = q_{2,1}$ (resp. $q = q_{2,2}$).

of estimator (5.11) is lower than the bias of the empirical estimator. In down and right of Figure 6.3, the upper bound obtained from a sequential strategy of simulation is more precise than the empirical quantile estimator. It is also illustrated that the initialisation step provides a more precise upper bound than the Monte Carlo-based estimator.

In down and right, supplementary information on the strategy of simulation is provided. First, a standard Monte Carlo approach does not provide a lower bound for q in this case. This can be easily explained by the geometric criterion used to build such lower bound. Recall that the set of points must verify the following constraint

$$\mu(\mathbb{V}^+(\mathbf{x}_1, \ldots, \mathbf{x}_n)) \geq 1 - p.$$

This constraint becomes more and more difficult to verify while p goes to 0 and d increases. Then, a design of experiments based on crude Monte Carlo seems not to be adapted to obtain such lower bounds.

Estimation of a $p = 10^{-5}$-quantile

As said before, there is no value of reference for the p-quantile with $p = 10^{-5}$. Then, the mean and the variance of the estimators are compared (Figure 6.4) as well as the deterministic bounds (Figure 6.5).

Since there is no reference value for q, the deterministic bounds are helpful to compare the estimations obtained. First, the upper bound obtained from a sequential framework is significantly lower than the one obtained from a Monte Carlo strategy. As for the $p = 10^{-3}$ case, a lower bound has not been obtained from the identically distributed sample. Moreover, from Figures 6.4 and 6.5, the empirical estimator is not informative since it is almost equal to its upper bound.

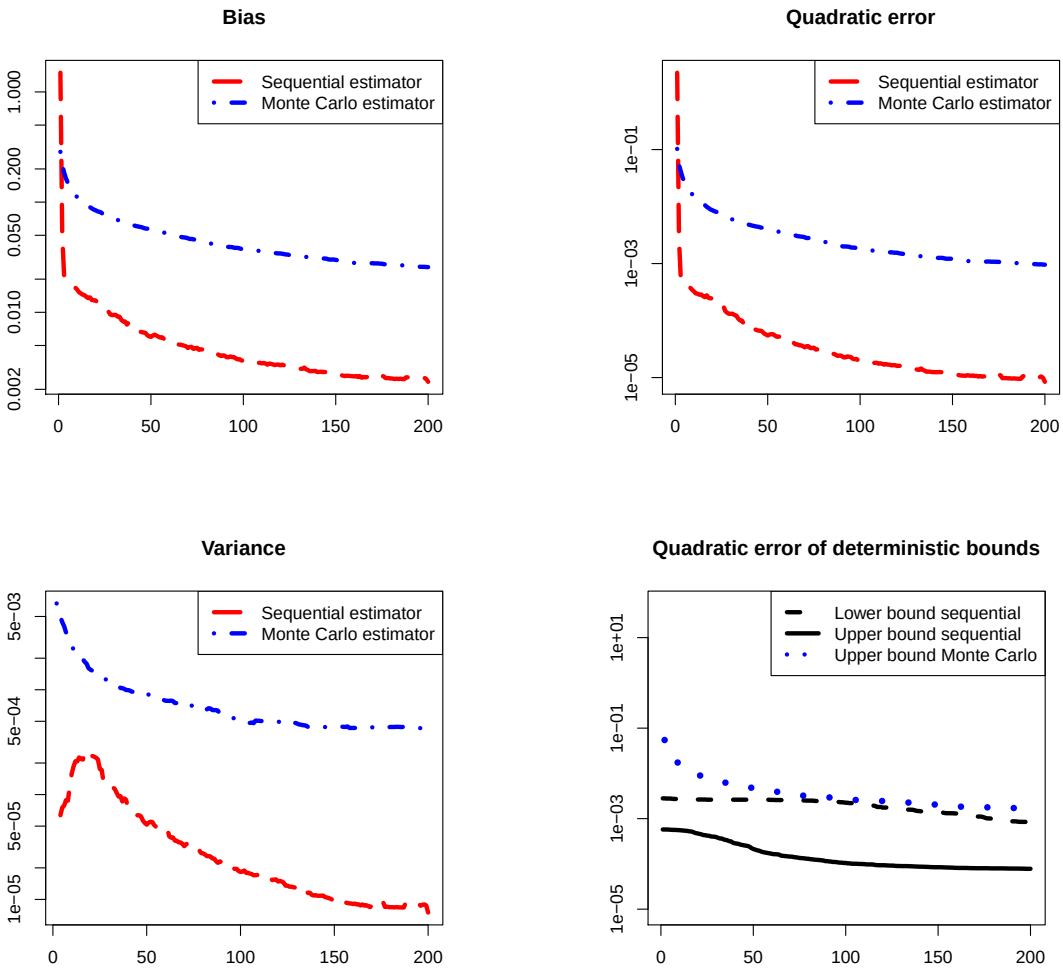


Figure 6.3: Industrial case for $p = 10^{-3}$ and $d = 2$. It is compared the bias, the quadratic error and the variance of the estimators in function of n. **Down-right:** Representation of the deterministic bounds obtained respectively from a sequential framework and a standard Monte Carlo sample. No lower bound has been obtained from a Monte Carlo sample.

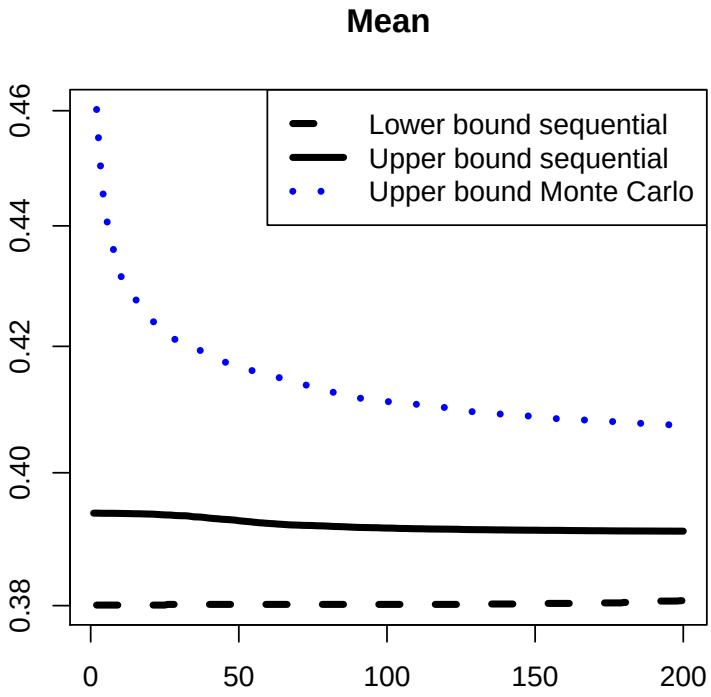


Figure 6.4: Industrial case for $p = 10^{-5}$ and $d = 2$. It is compared the mean of the deterministic bounds in function of n.

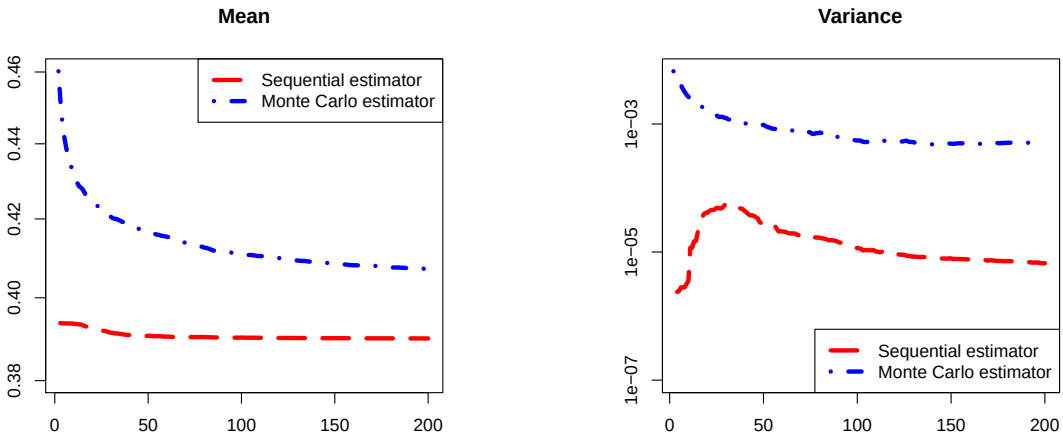


Figure 6.5: Industrial case for $p = 10^{-5}$ and $d = 2$. It is compared the mean and the variance of the estimators in function of n.

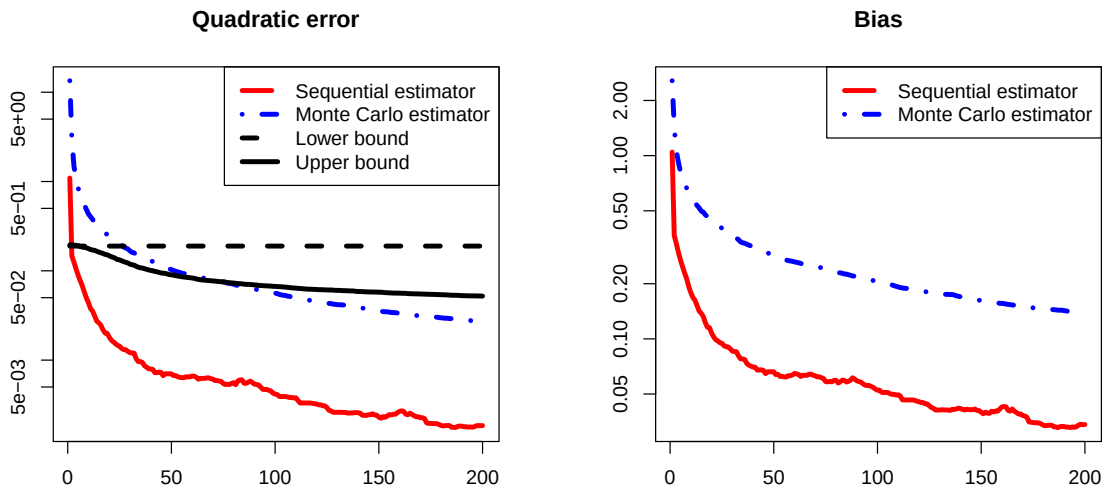


Figure 6.6: Real case study for $p = 10^{-3}$ and $d = 4$. The quadratic error (left) and the bias (right) is compared with the standard Monte Carlo and the sequential estimator in function of n. Results have been averaged on 100 independent experiments.

6.2.2 Four dimensional case

Estimation of a $p = 10^{-3}$-quantile

Figure 6.6 provides the quadratic error and the bias of the standard Monte Carlo and the sequential estimator in function of n. The sequential estimator outperforms the standard Monte Carlo estimator. The upper bound, represented by a plain line, has equivalent quadratic error than the Monte Carlo estimator. The estimation of the cumulative distribution function of $g(\mathbf{X})$ is represented in Figure 6.7. The curve in blue represents the empirical estimation of the cumulative distribution function obtained with a sample of size 10^5. The curve in red represents

$$\mathbb{E}\left[\widehat{F}_n(t)\right] = \frac{1}{n}\sum_{k=1}^{n} \mathbb{E}\left[p_{k-1}^- + (p_{k-1}^+ - p_{k-1}^-)\mathbb{1}_{\{g(\mathbf{X}_k)\leq t\}}\right],$$

for all $t \in [0,1]$ and $n = 200$. The two plain lines represent respectively $\mathbb{E}[q_n^-]$ and $\mathbb{E}[q_n^+]$. It must be noticed that $\widehat{F}_n(t)$ seems to be unbiased for $t \in [q_n^-, q_n^+]$. Since a quantile is deduced from the cumulative distribution function, all quantiles in $[q_n^-, q_n^+]$ can also be estimated.

Estimation of a $p = 10^{-5}$-quantile

Figure 6.8 provides the variance and the mean of the standard Monte Carlo and the sequential estimator in function of n. Figure 6.8b displays the values of $\mathbb{E}[\widehat{q}_n], \mathbb{E}[q_n^+]$ and $\mathbb{E}[\widehat{q}_n^{emp}]$. It must be noticed this comparison is not fair since 200 simulations is not sufficient to make a good estimation of a small quantile. Nonetheless, the upper bound is smaller than the empirical estimator. This indicates that a sequential sampling is more appropriate to bound a quantile for this numerical code. Results provided by Figure 6.8b has been normalised to be in $[0,1]$ as well as results represented in Figure 6.9. This figure illustrates the estimation of the cumulative distribution function of $g(\mathbf{X})$ between the bounds provided by the sequential framework. As for

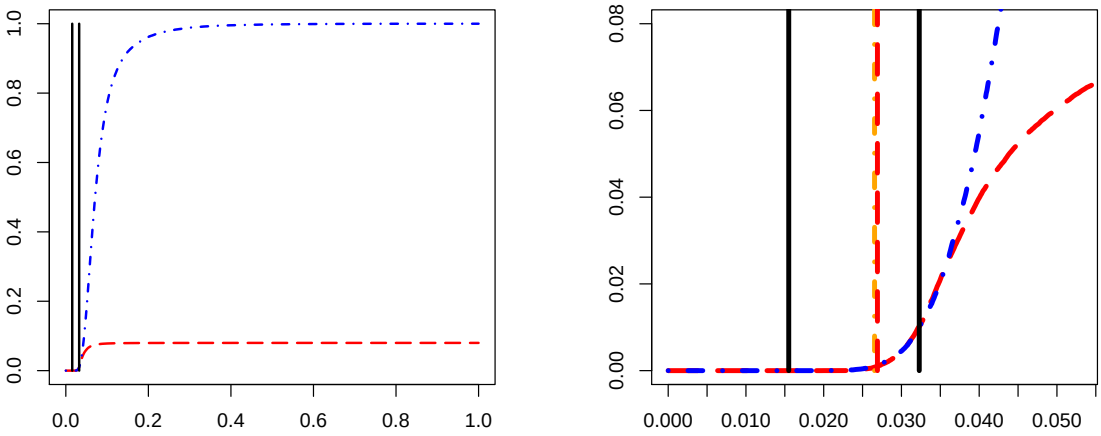
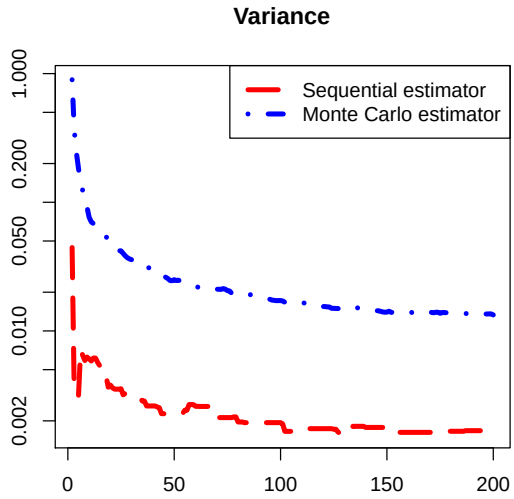
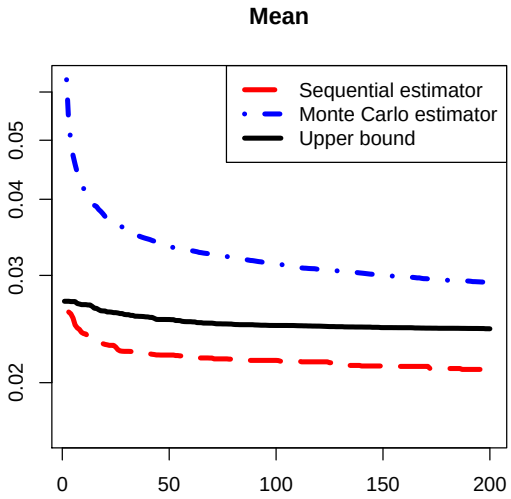


Figure 6.7: Real case study for $p = 10^{-3}$ and $d = 4$. Results have been averaged on 100 independent experiments. **Left:** Representation of the estimation of the cumulative distribution function of $g(\mathbf{X})$. Vertical lines represent the bounds obtained by the sequential estimator. **Right:** The same figure as in left but around the estimation of q_p. The red and orange dashed lines represent the estimation of the quantile respectively by Monte Carlo method and the sequential method. Results have been averaged on 100 independent experiments.

$p = 10^{-3}$, the sequential estimator seems to provide an unbiased estimator of the cumulative distribution function close to the deterministic bounds.

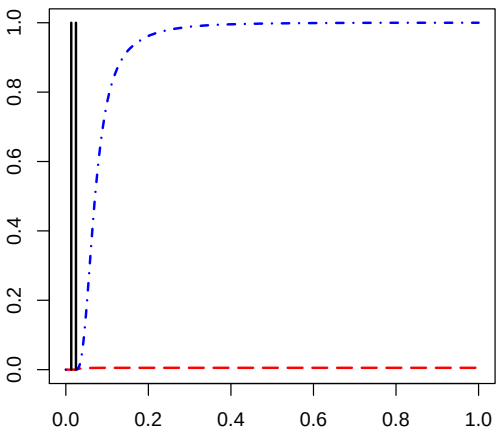


(a) Variance of the estimators in function of n.



(b) Mean of the estimators and the upper bound in function of n.

Figure 6.8: Real case study for $p = 10^{-5}$ and $d = 4$. Results have been averaged on 24 independent experiments.



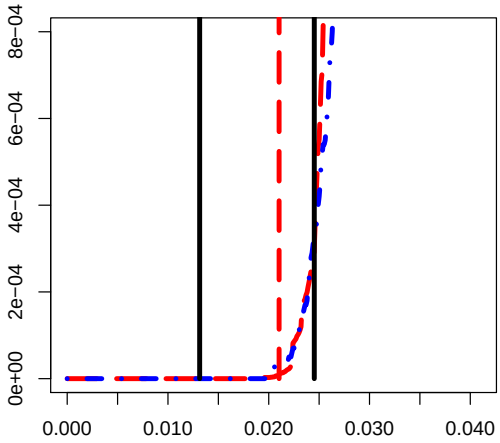


Figure 6.9: Real case study for $p = 10^{-5}$ and $d = 4$. Results have been averaged on 24 independent experiments. **Left:** Representation of the estimation of the cumulative distribution function of $g(\mathbf{X})$. Vertical lines represent the bounds obtained by the sequential estimator. **Right:** Representation of the estimation of the cumulative distribution function around the estimation of q_p. The red dashed line represents the estimation of the quantile obtained from the sequential method.

Conclusion

The first chapter of this thesis has provided a state of the art of probability estimation methods. In a first part, Monte Carlo methods are presented. Even if they do not suffer from the dimension they are not adapted for rare event probability estimation. Dedicated method for low probability estimation has been developed in an engineering context. FORM/SORM require to much hypotheses on the numerical code and since the estimator is deterministic it cannot be controlled. Nonetheless, it can be coupled with importance sampling techniques. In multilevel splitting, the searched probability can be expressed as a product of different quantities. In a standard context a sample must be produced at each step. This is not optimal in a monotonic context since each runs of the numerical code allow to update the non-dominated set.

In the second chapter classical methods for probability estimation have been adapted to the monotonic case. In general, such methods are not optimal since the information provided by the monotonic hypothesis is not totally used. It has been highlighted that each simulation must be drawn in the non-dominated set. Indeed, the deterministic bounds of the probability bring already all the information. However, sequential methods seem to be more efficient to estimate the probability of a rare event. Sequential importance sampling seems to be the most suitable method to estimate such probability. Indeed, at each evaluation by the numerical code the importance density must be updated. The use of a criterion can be useful to accelerate the convergence of the upper bound. Nonetheless, to couple such a criterion with importance sampling seems to produce an uncontrollable estimator.

The third chapter focuses on the deterministic bounds provided by the monotonic hypothesis. The condition to ensure their convergence has been established. Moreover, their rate of convergence has been studied for different strategies of simulation. As expected a sequential framework accelerates significantly such convergence. A monotonic binary classifier based on Support Vector Machines was constructed under the hypothesis that the numerical code is monotone and convex or concave. The main interest of this classifier is that it can be update sequentially. In a monotonic context, such property is particularly useful. A non-naive reject method has been developed to simulate uniformly in the non-dominated space. This is particularly helpful when the probability to be estimated becomes small.

In chapter four, a more general sequence of importance density has been built to estimate a probability. Moreover, the use of some criteria, used to reduce the upper deterministic bound, can be exploit to calibrate an importance density. Nonetheless, the estimator obtained from such densities could has very large variance. Indeed, these importance densities are too far, in some sense, to the initial distribution. Finally, a compromise must be made between the reduction of the upper deterministic bound and the possibility to have an estimator. The simplest sequential framework of simulation has been chosen. It is unbiased and the upper bound obtained from this

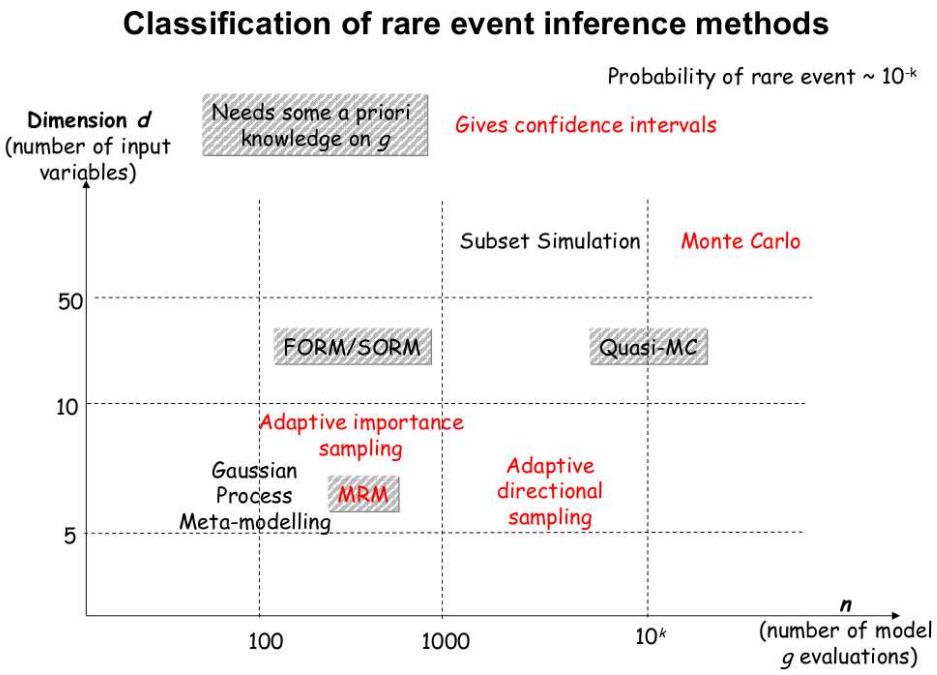


Figure 6.10: This figure is an update of Figure 1.8. Methods based on monotonicity hypothesis have been added.

framework and the optimal one are close. Unfortunately, it cannot be controlled by a central limit theorem.

The fifth chapter focuses on quantile estimation under monotonicity constraints. An initialisation step has been produced to determine a non-dominated set. This set is independent with the numerical code, it depends only on the value of p. Moreover, the information provided by this initialisation becomes more and more important while p becomes small. As for probability estimation, two deterministic bounds of a quantile can be obtained from a set of points. Using these two constructions, a sequential framework can be employed to estimate a quantile. Since a central limit for probability estimation is not available, such convergence result is also not available.

The main limit of monotonic hypothesis to estimate a probability and a quantile comes from the dimension. Indeed, the cost to compute the deterministic bounds increases exponentially with the dimension. But this cost depends also on the size of the sample. In practice the use of such methods implies that the total available number of evaluations by the numerical must be small. To represent these constraints Figure 6.10 compares such methods with the existing one. When the monotonic hypothesis is used, such class of methods are so-called Monotonic Reliability Method (MRM).

Perspectives

In chapter 3 the rate of convergence of the deterministic bounds in a sequential framework has not been deeply studied. A unsuccessful attempt to study such framework has been made.

Numerical studies have shown that this rate decreases while the dimension increases. There should be a dimension d such that the use of monotonic property is no more informative to bound a probability. The rate of convergence of the deterministic bounds of a quantile has not been studied. Since it depends on two algorithms, such rate seems to be difficult to obtain in practice.

Many thematics have not been studied in this thesis. When the dimension increases, there is no reason that the numerical code is still totally monotone. In this case, the lower and upper bounds are no longer available. Conditionally to the non-monotone inputs, the numerical code is monotone. Then, usual monotonic based method can be applied. It must be studied if a method inspiring by conditional Monte Carlo (see Section 1.5) is useful in practice.

Another theme is the influence of the input on a function of the numerical code. For example, sensitivity analysis to probability estimation has been recently developed in [78]. Since such methods require many evaluations by the numerical code, the use of the monotonic hypothesis can be useful.

In this thesis, it was assumed that the input are independent. It has not been studied the influence of a dependence relation and if the model can be also reduced to a globally increasing function.

Some of the methods provided in this thesis have or will be implemented in a package so-called MISTRAL coded in R and available on CRAN [98].

Bibliography

[1] Barry C Arnold, N Balakrishnan, and HN Nagaraja. A first course in order statistics (classics in applied mathematics). 2008.

[2] Bouhari Arouna. Adaptative Monte Carlo method, a variance reduction technique. *Monte Carlo Methods and Applications*, 10(1):1–24, 2004.

[3] Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

[4] Jean-Yves Audibert and Alexandre B Tsybakov. Fast learning rates for plug-in classifiers. *The Annals of statistics*, 35(2):608–633, 2007.

[5] Adrian Baddeley, Imre Bárány, and Rolf Schneider. Random polytopes, convex bodies, and approximation. *Stochastic Geometry: Lectures given at the CIME Summer School held in Martina Franca, Italy, September 13–18, 2004*, pages 77–118, 2007.

[6] Adrian Baddeley, Imre Bárány, and Rolf Schneider. Random polytopes, convex bodies, and approximation. *Stochastic Geometry: Lectures given at the CIME Summer School held in Martina Franca, Italy, September 13–18, 2004*, pages 77–118, 2007.

[7] I Bárány and DG Larman. Convex bodies, economic cap coverings, random polytopes. *Mathematika*, 35:274–291, 1988.

[8] Julien Bect, David Ginsbourger, Ling Li, Victor Picheny, and Emmanuel Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012.

[9] Julien Bect, Bousquet Nicolas, Bertrand Iooss, Shijie Liu, Alice Mabille, Anne-Laure Popelin, Thibault Riviere, Rémi Stroh, Roman Sueur, and Emmanuel Vazquez. Quantification et réduction de l'incertitude concernant les propriétés de monotonie d'un code de calcul coûteux à évaluer. In *JdS 2014*, 2014.

[10] Philippe Bekaert, Mateu Sbert, and Yves D Willems. *Weighted importance sampling techniques for Monte Carlo radiosity*. Springer, 2000.

[11] Bernard Bercu and Abderrahmen Touati. Exponential inequalities for self-normalized martingales with applications. *The Annals of Applied Probability*, 18(5):1848–1869, 2008.

[12] Alexandros Beskos, Ajay Jasra, and Alexandre Thiery. On the convergence of adaptive sequential Monte Carlo methods. *arXiv preprint arXiv:1306.6462*, 2013.

[13] Peter Bjerager. Probability integration by directional simulation. *Journal of Engineering Mechanics*, 114(8):1285–1302, 1988.

[14] Jerzy BŁaszczyński, Salvatore Greco, and Roman SŁowiński. Inductive discovery of laws using monotonic rules. *Engineering Applications of Artificial Intelligence*, 25(2):284–294, 2012.

[15] Jean-Marc Bourinet, François Deheeger, and Maurice Lemaire. Assessing small failure probabilities by combined subset simulation and support vector machines. *Structural Safety*, 33(6):343–353, 2011.

[16] Nicolas Bousquet. Accelerated Monte Carlo estimation of exceedance probabilities under monotonicity constraints. *Annales de la Faculté des Sciences de Toulouse*, 21:557–591, 2012.

[17] Nicolas Bousquet, Thierry Klein, and Vincent Moutoussamy. Approximation of limit state surfaces in monotonic Monte Carlo settings. *Submitted*, 2015.

[18] Karl Breitung. Asymptotic approximations for multinormal integrals. *Journal of Engineering Mechanics*, 110(3):357–366, 1984.

[19] Claire Cannamela. *Apport des méthodes probabilistes dans la simulation du comportement sous irradiation du combustible à particule*. Thèse de Doctorat, Université Paris 7, 2007.

[20] Claire Cannamela, Josselin Garnier, and Bertrand Iooss. Controlled stratification for quantile estimation. *The Annals of Applied Statistics*, pages 1554–1580, 2008.

[21] Olivier Cappé, Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P Robert. Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459, 2008.

[22] Alberto Rodríguez Casal. Set estimation under convexity type assumptions. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 43, pages 763–774, 2007.

[23] John L. Casti. *Would-be worlds: How simulation is changing the frontiers of science*. John Wiley and Sons, 1997.

[24] Frédéric Cérou, Pierre Del Moral, Teddy Furon, and Arnaud Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22(3):795–808, 2012.

[25] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, 2007.

[26] Yuguo Chen, Junyi Xie, and Jun S Liu. Stopping-time resampling for sequential Monte Carlo methods. *Quality control and applied statistics*, 51(1):39, 2006.

[27] Nicolas Chopin. Central limit theorem for sequential Monte Carlo methods and its application to bayesian inference. *Annals of Statistics*, pages 2385–2411, 2004.

[28] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

[29] William G. Cochran. *Sampling techniques*. John Wiley & Sons, 2007.

[30] Julien Cornebise, Éric Moulines, and Jimmy Olsson. Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18(4):461–480, 2008.

[31] Jean Cornuet, Jean-Michel Marin, Antonietta Mira, and Christian P. Robert. Adaptive multiple importance sampling. *Scandinavian Journal of Statistics*, 39(4):798–812, 2012.

[32] Antonio Cuevas and Ricardo Fraiman. A plug-in approach to support estimation. *The Annals of Statistics*, pages 2300–2312, 1997.

[33] Sébastien Da Veiga and Amandine Marrel. Gaussian process modeling with inequality constraints. In *Annales de la Faculté des Sciences de Toulouse*, volume 21, pages 529–555, 2012.

[34] Guillaume Damblin, Mathieu Couplet, and Bertrand Iooss. Numerical studies of space-filling designs: optimization of latin hypercube samples and subprojection properties. *Journal of Simulation*, 7(4):276–289, 2013.

[35] Hennie Daniels and Marina Velikova. Monotone and partially monotone neural networks. *Neural Networks, IEEE Transactions on*, 21(6):906–917, 2010.

[36] Herbert Aron David and Haikady Navada Nagaraja. *Order statistics*. Wiley Online Library, 1970.

[37] Scott De Marchi. *Computational and mathematical modeling in the social sciences*. Cambridge University Press, 2005.

[38] Etienne De Rocquigny. Structural reliability under monotony: Properties of form, simulation or response surface methods and a new class of monotonous reliability methods (mrm). *Structural Safety*, 31(5):363–374, 2009.

[39] François Deheeger. *Couplage mécano-fiabiliste: 2 SMART - méthodologie d'apprentissage stochastique en fiabilité*. Thèse de Doctorat, Université Blaise Pascal-Clermont-Ferrand II, 2008.

[40] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[41] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012.

[42] Armen Der Kiureghian and Taleen Dakessian. Multiple design points in first and second-order reliability. *Structural Safety*, 20(1):37–49, 1998.

[43] Ove Ditlevsen, Robert E; Melchers, and H. Gluver. General multi-dimensional probability integration by directional simulation. *Computers & Structures*, 36(2):355–368, 1990.

[44] Randal Douc, Arnaud Guillin, Jean-Michel Marin, and Christian P. Robert. Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics*, 11:427–447, 2007.

[45] Randal Douc and Eric Moulines. Limit theorems for weighted samples with applications to sequential Monte Carlo methods. In *ESAIM: Proceedings*, volume 19, pages 101–107. EDP Sciences, 2007.

[46] Michael Doumpos and Constantin Zopounidis. Monotonic support vector machines for credit risk rating. *New Mathematics and Natural Computation*, 5(03):557–570, 2009.

[47] Vincent Dubourg, François Deheeger, and Bruno Sudret. Metamodel-based importance sampling for the simulation of rare events. *Applications of Statistics and Probability in Civil Engineering*, 26:192, 2011.

[48] Lutz Dümbgen and Günther Walther. Rates of convergence for random approximations of convex sets. *Advances in applied probability*, pages 384–393, 1996.

[49] Cécile Durot. Monotone nonparametric regression with random design. *Mathematical Methods of Statistics*, 17(4):327–341, 2008.

[50] Anne Dutfoy and Régis Lebrun. Le test du maximum fort: une façon efficace de valider la qualité d'un point de conception. *18ème Congrès Français de Mécanique (Grenoble 2007)*, 2007.

[51] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994.

[52] Daniel Egloff, Markus Leippold, et al. Quantile estimation with adaptive importance sampling. *The Annals of Statistics*, 38(2):1244–1278, 2010.

[53] Kai-Tai Fang, Runze Li, and Agus Sudjianto. *Design and modeling for computer experiments*. Chapman, 2006.

[54] Paul Fearnhead, Benjamin M Taylor, et al. An adaptive sequential Monte Carlo sampler. *Bayesian Analysis*, 8(2):411–438, 2013.

[55] José Figueira, Salvatore Greco, and Matthias Ehrgott. *Multiple criteria decision analysis: state of the art surveys*, volume 78. Springer Science & Business Media, 2005.

[56] Sébastien Gadat, Thierry Klein, and Clément Marteau. Classification with the nearest neighbor rule in general finite dimensional spaces: necessary and sufficient conditions. *arXiv preprint arXiv:1411.0894*, 2014.

[57] Anne Gille-Genest. *Utilisation des méthodes numériques probabilistes dans les applications au domaine de la Fiabilité des Structures*. Thèse de Doctorat, Université Paris 6, 1999.

[58] Peter W. Glynn. Importance sampling for Monte Carlo estimation of quantiles. In *Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation*, pages 180–185, 1996.

[59] Shirin Golchi, Derek R Bingham, Hugh Chipman, and David A Campbell. Monotone function estimation for computer experiments. *arXiv preprint arXiv:1309.3802*, 2013.

[60] Donald Goldfarb and Ashok Idnani. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical programming*, 27(1):1–33, 1983.

[61] Arnaud Guyader, Nicolas Hengartner, and Eric Matzner-Løber. Simulation and estimation of extreme quantiles and extreme probabilities. *Applied Mathematics & Optimization*, 64(2):171–196, 2011.

[62] Peter Hall and Christopher C. Heyde. *Martingale limit theory and its application*. Academic press New York, 1980.

[63] John Michael Hammersley and David Christopher Handscomb. *Monte Carlo methods*, volume 1. Springer, 1964.

[64] Abraham M. Hasofer and Niels C. Lind. Exact and invariant second-moment code format. *Journal of the Engineering Mechanics division*, 100(1):111–121, 1974.

[65] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2008.

[66] Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194, 1995.

[67] Edmund Hlawka. Funktionen von beschränkter variatiou in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata*, 54(1):325–333, 1961.

[68] Jorge E. Hurtado. An examination of methods for approximating implicit limit state functions from the viewpoint of statistical learning theory. *Structural Safety*, 26(3):271–293, 2004.

[69] Jorge E Hurtado. Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis. *Structural Safety*, 29(1):2–15, 2007.

[70] Ruichen Jin, Wei Chen, and Agus Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134(1):268–287, 2005.

[71] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

[72] Benjamin Jourdain, Jérôme Lelong, et al. Robust adaptive importance sampling for normal random vectors. *The Annals of Applied Probability*, 19(5):1687–1718, 2009.

[73] Roger Koenker. *Quantile regression*. Cambridge university press, 2005.

[74] Agnes Lagnoux. Rare event simulation. *Probability Engineering Informatic Science*, 20:45–66, 2006.

[75] Agnes Lagnoux. Effective branching splitting method under cost constraint. *Stochastic Processes and their applications*, 118:1820–1851, 2008.

[76] François Le Gland and Nadia Oudjane. *A sequential particle algorithm that keeps the particle system alive*. Technical report, Rapport de recherche 5826, INRIA, 2006.

[77] Maurice Lemaire. *Structural reliability*, volume 84. John Wiley & Sons, 2010.

[78] Paul Lemaître. *Analyse de sensibilité en fiabilité des structures*. Thèse de Doctorat, Université Bordeaux 1, 2014.

[79] Christiane Lemieux. *Monte Carlo and Quasi-Monte Carlo sampling*. Springer, 2009.

[80] Faming Liang. Dynamically weighted importance sampling in Monte Carlo computation. *Journal of the American Statistical Association*, 97(459):807–821, 2002.

[81] Philipp Limbourg, Etienne De Rocquigny, and Guennadi Andrianov. Accelerated uncertainty propagation in two-level probabilistic studies under monotony. *Reliability Engineering & System Safety*, 95(9):998–1010, 2010.

[82] Jun S. Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[83] Michael D. McKay, Richard J. Beckman, and William J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

[84] R. E. Melchers. Importance sampling in structural systems. *Structural safety*, 6(1):3–10, 1989.

[85] Jérôme Morio. Extreme quantile estimation with nonparametric adaptive importance sampling. *Simulation Modelling Practice and Theory*, 27:76–89, 2012.

[86] Jérôme Morio, Mathieu Balesdent, Damien Jacquemart, and Christelle Vergé. A survey of rare event simulation methods for static input–output models. *Simulation Modelling Practice and Theory*, 49:287–304, 2014.

[87] Jérôme Morio, Rudy Pastel, and François Le Gland. Estimation de probabilités et de quantiles rares pour la caractérisation d'une zone de retombée d'un engin. *Journal de la Société Française de Statistique*, 152(4):1–29, 2012.

[88] Vincent Moutoussamy, Nicolas Bousquet, Bertrand Iooss, Paul Rochet, Thierry Klein, and Fabrice Gamboa. Comparing conservative estimations of failure probabilities using sequential designs of experiments in monotone frameworks. In *11th International Conference on Structural Safety & Reliability (ICOSSAR), June 16-20, 2013, New York*, 2013.

[89] Miguel Munoz Zuniga. *Méthodes stochastiques pour l'estimation contrôlée de faibles probabilités sur des modèles physiques complexes : application au domaine nucléaire*. Thèse de Doctorat, Université Paris 7, 2011.

[90] Miguel Munoz Zuniga, Josselin Garnier, Emmanuel Remy, and Etienne de Rocquigny. Adaptive directional stratification for controlled estimation of the probability of a rare event. *Reliability Engineering & System Safety*, 96(12):1691–1712, 2011.

[91] Harald Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(1):51–70, 1988.

[92] Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial Mathematics, 1992.

[93] Man-Suk Oh and James O. Berger. Adaptive importance sampling in Monte Carlo integration. *Journal of Statistical Computation and Simulation*, 41(3-4):143–168, 1992.

[94] Eric Parent and Jacques Bernier. *Le raisonnement bayésien. Modélisation et inférence*. Springer, 2007.

[95] Rudy Pastel. *Estimation de probabilités d'évènements rares et de quantiles extrêmes: applications dans le domaine aérospatial*. PhD thesis, Rennes 1, 2012.

[96] Kristiaan Pelckmans, Marcelo Espinoza, Jos De Brabanter, Johan AK Suykens, and Bart De Moor. Primal-dual monotone kernel regression. *Neural Processing Letters*, 22(2):171–182, 2005.

[97] M. Powell and J. Swann. Weighted uniform sampling—a Monte Carlo technique for reducing variance. *IMA Journal of Applied Mathematics*, 2(3):228–236, 1966.

[98] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015.

[99] Rüdiger Rackwitz and Bernd Flessler. Structural reliability under combined random load sequences. *Computers & Structures*, 9(5):489–494, 1978.

[100] Mohammadreza Rajabalinejad, LE Meester, PHAJM Van Gelder, and JK Vrijling. Dynamic bounds coupled with monte carlo simulations. *Reliability Engineering & System Safety*, 96(2):278–285, 2011.

[101] R. Tyrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1997.

[102] Gerardo Rubino, Bruno Tuffin, et al. *Rare event simulation using Monte Carlo methods*. Wiley Online Library, 2009.

[103] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo method*, volume 707. Wiley. com, 2011.

[104] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical science*, pages 409–423, 1989.

[105] Thomas J. Santner, Brian J. Williams, and William. I. Notz. *The design and analysis of computer experiments*. Springer, 2003.

[106] Amandine Schreck, Gersende Fort, Eric Moulines, and Matti Vihola. Convergence of markovian stochastic approximation with discontinuous dynamics. *arXiv preprint arXiv:1403.6803*, 2014.

[107] Thomas S. Shively, Thomas W. Sager, and Stephen G. Walker. A bayesian approach to non-parametric monotone function estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(1):159–175, 2009.

[108] Ralph C. Smith. *Uncertainty Quantification: Theory, Implementation, and Applications*, volume 12. SIAM, 2013.

[109] Ilya M. Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational mathematics and mathematical physics*, (7):86–112, 1967.

[110] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 2. Wiley New York, 1998.

[111] Marina Velikova, Hennie Daniels, and Ad Feelders. Solving partially monotone problems with neural networks. In *Proceedings of the International Conference on Neural Networks, Vienna, Austria*, 2006.

[112] Guenther Walther. On a generalization of blaschke's rolling theorem and the smoothing of surfaces. *Mathematical methods in the applied sciences*, 22(4):301–316, 1999.

[113] Yanling Zhang and A. Der Kiureghian. Two improved algorithms for reliability analysis. In *Reliability and optimization of structural systems*, pages 297–304. Springer, 1995.

Résumé

Cette thèse se place dans le contexte de la fiabilité structurale associée à des modèles numériques représentant un phénomène physique. On considère que la fiabilité est représentée par des indicateurs qui prennent la forme d'une probabilité et d'un quantile. Les modèles numériques étudiés sont considérés déterministes et de type boîte-noire. La connaissance du phénomène physique modélisé permet néanmoins de faire des hypothèses de forme sur ce modèle. La prise en compte des propriétés de monotonie dans l'établissement des indicateurs de risques constitue l'originalité de ce travail de thèse. Le principal intérêt de cette hypothèse est de pouvoir contrôler de façon certaine ces indicateurs. Ce contrôle prend la forme de bornes obtenues par le choix d'un plan d'expériences approprié. Les travaux de cette thèse se concentrent sur deux thématiques associées à cette hypothèse de monotonie. La première est l'étude de ces bornes pour l'estimation de probabilité. L'influence de la dimension et du plan d'expériences utilisé sur la qualité de l'encadrement pouvant mener à la dégradation d'un composant ou d'une structure industrielle sont étudiées. La seconde est de tirer parti de l'information de ces bornes pour estimer au mieux une probabilité ou un quantile. Pour l'estimation de probabilité, l'objectif est d'améliorer les méthodes existantes spécifiques à l'estimation de probabilité sous des contraintes de monotonie. Les principales étapes d'estimation de probabilité ont ensuite été adaptées à l'encadrement et l'estimation d'un quantile. Ces méthodes ont ensuite été mises en pratique sur un cas industriel.

Mots-clefs Apprentissage séquentiel; Expérience numériques; Fiabilité; Incertitudes; Monotonie; Probabilité; Quantile

Abstract

This thesis takes place in a structural reliability context which involves numerical model implementing a physical phenomenon. The reliability of an industrial component is summarised by two indicators of failure,a probability and a quantile. The studied numerical models are considered deterministic and black-box. Nonetheless, the knowledge of the studied physical phenomenon allows to make some hypothesis on this model. The original work of this thesis comes from considering monotonicity properties of the phenomenon for computing these indicators. The main interest of this hypothesis is to provide a sure control on these indicators. This control takes the form of bounds obtained by an appropriate design of numerical experiments. This thesis focuses on two themes associated to this monotonicity hypothesis. The first one is the study of these bounds for probability estimation. The influence of the dimension and the chosen design of experiments on the bounds are studied. The second one takes into account the information provided by these bounds to estimate as best as possible a probability or a quantile. For probability estimation, the aim is to improve the existing methods devoted to probability estimation under monotonicity constraints. The main steps built for probability estimation are then adapted to bound and estimate a quantile. These methods have then been applied on an industrial case.

Keywords Sequential learning; Computer experiments; Reliability; Uncertainties; Monotonicity; Probability; Quantile