



# Approximation and model reduction for partial differential equations with probabilistic interpretation

Arthur Macherey

## ► To cite this version:

Arthur Macherey. Approximation and model reduction for partial differential equations with probabilistic interpretation. Analysis of PDEs [math.AP]. École centrale de Nantes, 2021. English. NNT : 2021ECDN0026 . tel-03351933

**HAL Id: tel-03351933**

**<https://theses.hal.science/tel-03351933>**

Submitted on 22 Sep 2021

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

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

# THÈSE DE DOCTORAT DE

L'ÉCOLE CENTRALE DE NANTES

ÉCOLE DOCTORALE N° 601

*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*

*Spécialité : Mathématiques et leurs interactions*

Par

**Arthur MACHEREY**

**Approximation et réduction de modèle pour les équations aux  
dérivées partielles avec interprétation probabiliste**

Thèse présentée et soutenue à Centrale Nantes, le 28 juin 2021.

Unité de recherche : UMR\_C 6629, Laboratoire de Mathématiques Jean Leray (LMJL), Nantes

## Rapporteurs avant soutenance :

Mireille BOSSY Directrice de recherche, Université Côte d'Azur  
Benjamin JOURDAIN Professeur, École des Ponts ParisTech, Marne-la-Vallée

## Composition du Jury :

Président :	Tony LELIÈVRE	Professeur, École des Ponts ParisTech, Marne-la-Vallée
Examineurs :	Pierre ÉTORÉ	Maître de Conférences HDR, Grenoble INP
Dir. de thèse :	Anthony NOUY	Professeur des Universités, École Centrale de Nantes
Co-dir. de thèse :	Clémentine PRIEUR	Professeure des Universités, Université Grenoble Alpes
Co-enc. de thèse :	Marie BILLAUD-FRIESS	Maître de Conférences, École Centrale de Nantes



# REMERCIEMENTS

---

Je tiens tout d'abord à remercier vivement Mireille Bossy et Benjamin Jourdain d'avoir accepté de rapporter ma thèse. Merci pour vos relectures attentives de mon manuscrit. Merci également à Pierre Etoré et Tony Lelièvre pour avoir accepté d'être les examinateurs de ma thèse. Merci en particulier à ce dernier d'avoir accepté de présider le jury. Merci à tou.te.s pour vos remarques et vos questions suite à la soutenance, j'ai pris un grand plaisir à échanger avec vous sur mes sujets d'étude. Espérons juste que la prochaine fois ces échanges se feront en face à face et pas derrière nos écrans.

Ensuite j'aimerais remercier mes encadrant.e.s de thèse. Tout d'abord Clémentine. Merci de ton accueil à Grenoble, dans un cadre vraiment propice au travail ... et aux activités de montagne diverses et variées que j'ai eu l'opportunité de pratiquer dans les massifs alentours. Merci ensuite à Marie pour ton encadrement bienveillant et ta grande disponibilité, que ce soit pour parler de différents aspects pratiques et théoriques de ma thèse ou pour m'aider à soumettre mon premier papier un 1er janvier. Enfin merci à Anthony. La liste des choses que je te dois est trop longue pour figurer intégralement ici, donc je vais aller à l'essentiel. Tout d'abord merci pour toutes les connaissances scientifiques que tu m'as partagées tout au long de ces trois années. Merci du temps que tu m'as accordé, pour écouter et répondre à mes questions, qu'elles soient scientifiques ou plus personnelles. Merci de m'avoir montré ce qu'est un chercheur passionné et impliqué, notamment auprès de ses doctorants. Encore merci à vous trois, vous avez su, chacun.e à votre manière m'apporter ce qui m'a permis d'aller au bout de ce travail.

Enfin si mes encadrant ;e.s ont largement contribué à l'aboutissement de cette thèse, ils ne sont pas les seul.e.s à m'avoir soutenu. Et sans ces autres personnes je ne serai pas arrivé à bon port aussi "facilement". Merci donc à toutes les personnes qui m'ont soutenu, de près ou de loin, au cours de ma thèse.

Je pense tout d'abord à la frange grenobloise des soutiens. Merci à l'équipe Airsea pour son accueil au top, moi qui débarquais à Grenoble sans y connaître personne. Merci

en particulier à Annie pour tout le soutien administratif. Réaliser une thèse à cheval entre deux lieux géographiques n'est pas toujours évident mais avec ton aide ça n'a jamais paru très compliqué. Bien sûr merci à mes co-bureau. Que ce soit Reda, fidèle au poste depuis la première heure et toujours disponible pour un "bon café", un match de foot le midi ou une bière en soirée. Ou Philo qui nous a rejoint en cours de route et avec qui les discussions, scientifiques évidemment, auront été au moins aussi animées que les soirées. Ou Mathieu pour quelques mois, lui qui a bien participé aux discussions "scientifiques" du bureau. Merci à tou.te.s les trois pour votre soutien, je sais qu'on se recroisera très bientôt pour fêter ma thèse, celle de Philo ou, je l'espère, la chute du système capitaliste actuel. Merci également aux autres doctorant.e.s du labo et de l'équipe, mention toute spéciale pour Sophie et Victor le mangeur de fleur ... votre bureau était bien mais le nôtre était mieux. Merci aussi à Olivier pour avoir encore resserré les liens entre permanent.e.s et doctorant.e.s dans l'équipe, liens déjà savamment entretenus par nos séminaires d'équipe très intéressants à tous les points de vue.

Merci bien sûr à tou.te.s les membres du LJK que j'ai côtoyé au cours de mes quelques mois grenoblois. Merci en particulier à Laurence, Fred et Cathy pour les coups de main (informatiques, administratifs, mais pas que). Dédicace spéciale aux 7 nains pour les séances de course à pied intenses, suivies de soirées encore plus intenses et de lendemain beaucoup plus compliqués (sauf pour Francky, ce qui restera toujours un mystère).

J'en termine par les ami.e.s de Grenoble, toujours là pour un ravitaillement œnologique ou zytologique en soirée ou un aller-retour en montagne pour se changer les idées (ou se perdre pour les plus chanceux). Je pense bien sûr à Pablo & Milena, Francine & Yann, Félix, Marielle et Vincent.

Viens la frange nantaise des soutiens, présente en nombre.

Tout d'abord merci à tou.te.s les membres du LMJL pour les échanges que nous avons pu avoir. Remerciements spéciaux pour Bénédicte et Alexandra pour leur soutien administratif et logistique au top. Quand on arrive en cours d'année dans un laboratoire, ce sont des soutiens qui comptent. Merci bien sûr aux co-bureau du bureau 013. Merci à Claire pour m'avoir gentiment accueilli et pour toutes les discussions sur l'après-thèse et les encouragements prodigués. Je suis content que tu aies réussi à trouver ta voie dans

le monde académique, s'il y a bien quelqu'un qui le mérite, c'est toi. Merci à Mazen de m'avoir rejoint ces derniers mois, c'est toujours agréable une personne dans le bureau pour ne pas péter un câble tout seul, surtout en période de rédaction. Et je ne désespère pas que tu trouves un jour un boulot à la hauteur de tes capacités en recherche ... et que tu élèves ta sensibilité politico-socio-environnementales à de telles hauteurs.

Merci ensuite à tou.te.s les doctorant.e.s du LMJL. Les ancien.ne.s d'abord avec Caro, Solène, Matthieu, Germain, Maha, Zeinab, Fabien et Alexandre. Merci pour les bons moments passés en conférence, à une après-midi des doctorants ou au labo. Les derniers n'ont pas été aussi nombreux que je voulais, à cause de mes nombreux aller-retours à Centrale, mais je garderai en tête votre superbe accueil et j'espère qu'on se recroisera à d'autres occasions. Merci aussi aux autres, rencontré.e.s plus récemment, et notamment Adrien, Adrian, Antoine, Honorine et Jérôme. Merci pour les échanges à distance en période de covid et pour les quelques bières qu'on a eu le temps d'aller boire entre les différents confinements. Les deux dernier.e.s, vous récupérez le bureau 013, faites-en bon usage.

Merci aussi aux centralien.ne.s du labo et au département inof-maths dans son ensemble. Plus particulièrement, merci à Erwan et Cécile mon grand frère et ma grande soeur de thèse. Merci pour votre accueil, vos conseils et vos disponibilités quand je suis revenu sur Nantes et après. Merci aussi à Flavien, Martin et Léa pour les bons moments passés quand j'étais à Centrale. Mention spéciale à Anthony, rencontre récente mais heureuse. Bon courage pour la fin de ta thèse, je sais qu'on va vite se recroiser.

Enfin les copains et copines de Nantes. Bon je vous aurai saoulé avec ma thèse ces dernières années, maintenant c'est bon c'est fini, promis on en parle plus. Merci à Étienne pour le soutien actif et les coups de pied aux fesses quand il le fallait. Merci aussi de m'avoir montré la voie, quand j'ai vu que tu pouvais avoir une thèse, mon syndrome de l'imposteur a disparu. Merci aux autres pour les nombreux verres (d'eau) de soutien et séance intensives (mais pas très régulières pour moi) de course à pied et de vélo, en particulier Raphaëlle, Raphaël, Vincent, Constance et Ludovic. Petite dédicace à Gildas aussi, maintenant que je vais avoir du temps pour m'entraîner, plus question de perdre au squash. Et puis vu qu'on approche de la fin, il faut mentionner les meilleur.e.s. Thomas. Merci pour tout ... les dépannages d'ordinateur quand le mien m'a lâché, les (trop) nombreux verres de soutien et les coups de pied aux fesses salvateurs. Clairement si j'ai

fini c'est grâce à ta phrase : "tu pourras aller dire à tous ceux qui disent de la merde dans les médias que tu viens faire pareil mais qu'en plus tu as une thèse". Enfin Marion. La collocataire parfaite avec qui faire une thèse. Franchement malgré les périodes pas faciles qu'on a traversées tous les deux (deux thésard.e.s en rédaction dans le même appartement et en temps de covid ça en fait des raisons de péter des câbles) et malgré les confinements multiples on ne s'est jamais entretenu, et ça c'est grâce à ta tolérance (sisi) à toute épreuve. Merci pour toutes les discussions tard le soir (pas trop tôt le matin, bizarre) qu'on a pu avoir, sur tout et n'importe quoi et pour ton soutien aussi bien mental que logistique (on m'a soufflé que soutien physique pouvait être mal interprété donc j'ai changé). Pas sûr que j'y sois arrivé sans toi et le (brillant) exemple que tu m'as montré (en plus du soutien important sus-mentionné).

Une dernière catégorie de remerciements : les inclassables. A ma famille tout d'abord, merci pour votre soutien tout au long de ces trois années. En particulier à ma maman pour m'avoir poussé sans rien comprendre à mon sujet. A mon papa aussi pour avoir été persuadé toutes ces années qu'il pourrait comprendre ce que je fais en enseignant les maths dans le secondaire ... je ne pense pas avoir vulgarisé mon sujet aussi bien que tu le fais avec les maths en classe. Mais clairement si je me suis orienté dans les maths et vers l'enseignement je te le dois (au moins en partie). Merci aussi à mes sœurs qui ont eu des échéances importantes en même temps que moi. Je n'ai pas été trop là pour vous ces derniers temps, mais promis je vais me rattraper. Et finalement merci à mes grand-mères pour leur soutien même si "faire de la recherche" c'est pas tous les jours très clair et puis après tout "c'est quand que tu fais un vrai métier?". Les oncles, tantes et cousin.e.s ne sont pas oublié.e.s, merci pour votre soutien, quelle que soit la forme qu'il ait pris.

Il y a aussi les copains qui ont suivi ça de près ou de loin. Hélène tout d'abord pour les multiples lettres de soutien échangées. Tes prises de recul m'auront été salutaires (enfin elles l'auront été pour que je finisse ma thèse). Merci pour les discussions sur l'après-thèse aussi, qui m'ont donné envie de finir vite pour me concentrer pleinement à l'après. Merci aux Sportings aussi, je l'ai finie ma thèse Roland-Garros ... et tout ça presque sans louper un match. Merci aux zouzous de Besançon et de Neuville, les (trop rares) petites parties de pêche, séances de jardinage ou soirées au cours de ces 3 dernières années ont toujours été un agréable bol d'air et un bon moyen de couper avec la thèse. Il s'en est passé des choses ces trois dernières années, maintenant on va avoir le temps de fêter ça.

Et une Petit(e) dernière pour la fin, toi qui a su m'apporter ton soutien et tes encouragements au cours des derniers mois passés ensemble et ce malgré tes échéances synchronisées avec les miennes. Merci Camille. Je suis content d'avoir désormais un peu de temps rien qu'à nous, libéré du poids de nos projets professionnels respectifs.





# Contents

List of Figures . . . . .	11
List of Tables . . . . .	12
List of Algorithms . . . . .	13
<b>Introduction (version française)</b>	<b>15</b>
<b>Introduction (english version)</b>	<b>21</b>
<b>1 Probabilistic Methods for PDEs</b>	<b>27</b>
1.1 Stochastic Differential Equations . . . . .	29
1.2 Feynman-Kac Representation . . . . .	31
1.2.1 Backward Parabolic Boundary Value Problem . . . . .	32
1.2.2 Elliptic Boundary Value Problem . . . . .	33
1.3 Numerical Methods . . . . .	34
1.3.1 Approximation of Diffusion Processes . . . . .	35
1.3.2 Monte-Carlo Methods . . . . .	35
1.3.3 Statistical Error for the Simulation of Pointwise Evaluations . . . . .	36
<b>2 Probabilistic Methods for Solving High-Dimensional Elliptic PDEs</b>	<b>45</b>
2.1 Introduction . . . . .	46
2.2 Probabilistic Tools for Solving PDEs . . . . .	47
2.2.1 Pointwise Evaluations of the Solution . . . . .	47
2.2.2 A Sequential Control Variates Algorithm . . . . .	50
2.3 Adaptive Sparse Interpolation . . . . .	52
2.3.1 Sparse Interpolation . . . . .	52
2.3.2 Adaptive Algorithm for Sparse Interpolation . . . . .	53
2.4 Main Contributions . . . . .	54
2.4.1 Perturbed Version of Sparse Interpolation . . . . .	54
2.4.2 Adaptive Version of Sequential Control Variate Algorithm . . . . .	54
2.4.3 Numerical Results . . . . .	55
2.5 Conclusion . . . . .	59
<b>3 PAC Algorithm in Relative Precision</b>	<b>61</b>
3.1 Introduction . . . . .	62
3.2 Monte-Carlo Estimate with Guaranteed Relative Precision . . . . .	63
3.2.1 Monte-Carlo Estimate . . . . .	64
3.2.2 Complexity Analysis . . . . .	66
3.3 Optimization Algorithms with Guaranteed Relative Precision . . . . .	66
3.3.1 Non-Adaptive Algorithm . . . . .	67
3.3.2 Adaptive Algorithm . . . . .	69
3.4 Numerical Results . . . . .	71

3.A	Intermediate Results . . . . .	75
3.B	Proof of Proposition 3.4 . . . . .	77
<b>4</b>	<b>Reduced Basis Methods</b>	<b>81</b>
4.1	Idealized RBM . . . . .	82
4.1.1	Solution Manifold and Kolmogorov Width . . . . .	82
4.1.2	Greedy Algorithm . . . . .	83
4.1.3	Snapshot-Based Reduced Spaces . . . . .	84
4.2	Feasible Version of RBM . . . . .	85
4.2.1	Truth Approximation . . . . .	85
4.2.2	Feasible Projection . . . . .	86
4.2.3	Discrete Training Set . . . . .	86
4.2.4	Practical Greedy Algorithm . . . . .	88
4.3	RBM for Parameter-Dependent PDEs . . . . .	89
4.3.1	Weak-Formulation . . . . .	89
4.3.2	Affine Decomposition and EIM . . . . .	90
4.3.3	Error Estimation . . . . .	90
<b>5</b>	<b>Probabilistic Reduced Basis Methods</b>	<b>93</b>
5.1	Probabilistic Representation for Parameter-Dependent PDEs . . . . .	94
5.1.1	Elliptic Parameter-Dependent PDEs . . . . .	94
5.1.2	Parabolic Parameter-Dependent PDEs . . . . .	95
5.2	Error Estimation using Probabilistic Representation . . . . .	96
5.2.1	Parameter-Dependent Elliptic PDEs . . . . .	97
5.2.2	Parameter-Dependent Parabolic PDEs . . . . .	98
5.2.3	Probably Approximately Correct Algorithm . . . . .	99
5.3	Probabilistic Greedy Algorithms . . . . .	100
5.3.1	With Probabilistic Error Estimate . . . . .	100
5.3.2	With Random Training Sets . . . . .	102
	<b>Conclusion and Outlook</b>	<b>107</b>

# List of Figures

1.1	Comparison between the estimation of the exit time $\tau^{t,x,\Delta t}$ of $\mathcal{D}$ and the estimation of the exit time $\tau_1^{t,x,\Delta t}$ of $\mathcal{D}^{\Delta t}$ [50], illustrated for a particular trajectory of $X^{t,x,\Delta t}$ . The domain $\mathcal{D}^{\Delta t}$ (delimited by - - -) allows to estimate an exit position (and exit time) with a bias lower than the one using $\mathcal{D}$ (delimited by —). . . . .	39
1.2	Comparison between Walk on Sphere (left) and Green Function First Passage (right) techniques [62]. . . . .	39
2.1	(TC1) Algorithm 2.1 for fixed $\Lambda$ : evolution of $\ u - \tilde{u}_{\Lambda^6}^k\ $ with respect to $k$ for various $M$ (left figure), and various $\Delta t$ (right figure). . . . .	55
2.2	(TC1) Algorithm 2.1 for fixed $\Lambda^i$ : evolution of $\ u - \tilde{u}_{\Lambda^i}^k\ _2$ with respect to $k$ for $i = 8$ (left figure), and $i = 10$ (right figure). . . . .	56
2.3	(TC1) Comparison of Algorithm 2.2 applied to exact solution and Algorithm 2.3 : (left) absolute error in $L^2$ -norm (right) evolution of $\varepsilon^n$ and $\tilde{\varepsilon}^n$ with respect to $\#\Lambda_n$ . . . . .	57
2.4	(TC1) Adaptive Algorithm 2.1: evolution of $\ u - u_{\Lambda_k}^k\ _2$ (continuous line) and $\ u - u_{\Lambda_k}^k\ _\infty$ (dashed line) with respect to step $k$ and $\Delta t$ . . .	58
3.1	Final state of each algorithm after one run with $\tau = 0.1$ , $\lambda = 0.1$ and $\tau_{abs} = \tau \mathbb{E}[Z(\xi^*)] $ for ME Algorithm. Left scale : values of the estimates $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$ together with the associated confidence intervals of level $1 - d_{m(\xi)}$ . Right scale : values of $m(\xi)$ . . . . .	72
3.2	Evolution of $\Xi_n$ and number of samples $m_n(\xi)$ with $n$ for Algorithm 3.2 with $\tau = \lambda = 0.1$ . . . . .	74
3.3	Average complexity $\mathbb{E}(\mathcal{M})$ of Algorithm 3.2 with respect to $\tau$ and $\lambda$ (in log-log scale). . . . .	75

# List of Tables

1.1	Summary of the advantages and drawbacks of each numerical time-integration technique. . . . .	41
2.1	Algorithm 2.2 computed on the exact solution of (TC1): evolution of $\#\Lambda_n$ , error criterion $\varepsilon^n$ and interpolation errors in norms $L^2$ and $L^\infty$ at each step $n$ . . . . .	56
2.2	(TC1) Comparison of the algorithmic complexity to reach the precision $3 \cdot 10^{-5}$ , with $(\Delta t, M) = (10^{-4}, 1000)$ . . . . .	57
2.3	(TC2) Comparison of Algorithm 2.2 (first four columns) and Algorithm 2.3 (last four columns). . . . .	59
2.4	(TC3) Comparison of Algorithm 2.2 (first four columns) and Algorithm 2.3 (last four columns). . . . .	59
3.1	Average complexity $\mathbb{E}(\mathcal{M})$ , estimated using 30 runs for each algorithm, with $\tau = 0.1$ , $\lambda = 0.1$ and $\tau_{abs} = \tau \mathbb{E}[Z(\xi^*)] $ for ME algorithm. . . . .	73
3.2	Estimated runtime $T$ (in seconds) for different values of $t^*$ , with $\tau = \lambda = 0.1$ and $\tau_{abs} = \tau \mathbb{E}[Z(\xi^*)] $ for ME algorithm. . . . .	74

# List of Algorithms

2.1	Sequential Control Variates Algorithm [49]	50
2.2	Adaptive Interpolation Algorithm [22]	53
2.3	Perturbed adaptive sparse interpolation algorithm	54
3.1	Non-Adaptive PAC Algorithm in Relative Precision	67
3.2	Adaptive PAC Algorithm in Relative Precision	70
4.1	Greedy Algorithm for Reduced Basis Methods [32]	83
4.2	Practical Greedy Algorithm for RBM	88
5.1	Probabilistic Greedy Algorithm for RBM	100
5.2	Greedy Algorithm with Random Training Sets [27]	102
5.3	Full Probabilistic Greedy Algorithm	103



# Introduction (version française)

Cette thèse est consacrée à l'approximation de la solution d'équations aux dérivées partielles (EDPs) en grande dimension, éventuellement paramétrées. La résolution d'EDPs est un domaine très vaste et très ancien des mathématiques : les EDPs ont été analysées depuis la fin du 17<sup>ème</sup> siècle par Newton et Leibniz. Alors qu'à cette époque, seuls des modèles simples avec des solutions analytiques étaient étudiés, les récents développements en informatique ont ouvert la possibilité d'aborder des EDPs plus compliquées par le biais de méthodes numériques. Plus récemment, les EDPs dépendant des paramètres ont été considérées pour étudier des problèmes plus complexes en optimisation, contrôle, analyse de sensibilité ou quantification d'incertitude.

La solution d'une EDP est une fonction multivariée  $u(x)$  dépendant de certaines variables physiques (spatiales, temporelles)  $x$  dans  $\mathbb{R}^d$ . Pour les EDPs paramétrées, la solution  $u(x; y)$  dépend de variables supplémentaires  $y \in Y \subset \mathbb{R}^p$ , appelées paramètres.

Les problèmes concrets pour lesquels les EDPs sont utilisées proviennent de différents domaines, par exemple en physique, en biologie ou en finance. Ces applications nécessitent généralement de pouvoir approcher avec une certaine précision la solution  $u$  d'une EDP. Pour les EDPs paramétrées, des approximations précises de  $u(\cdot; y)$  pour de nombreuses valeurs  $y$  des paramètres sont généralement requises. Des approximations précises signifient généralement des modèles numériques fins, donc coûteux.

De plus, la complexité numérique de l'approximation des fonctions  $u(x)$  ou  $u(x; y)$  peut augmenter de manière exponentielle avec les dimensions  $d$  ou  $p$ . C'est ce qu'on appelle la malédiction de la dimension [88, 23]. Cette malédiction peut parfois être évitée, par exemple en utilisant les méthodes de Monte-Carlo pour l'estimation d'intégrales [85]. Lorsqu'il s'agit d'approximation de fonctions, qui est un problème plus difficile qu'un problème d'intégration, il n'est pas toujours possible d'éviter cette malédiction en utilisant des méthodes aléatoires [72].

Les EDPs en grande dimension donnent donc lieu à des problèmes d'approximation difficiles, aussi bien dans des cas indépendants des paramètres que dans des cas paramétrés. Pour les EDPs paramétrées, une solution consiste à utiliser un méta-modèle, également appelé modèle de substitution, qui approche  $u$  et dont l'évaluation est peu coûteuse par rapport au coût d'évaluation du modèle numérique initial. Il est particulièrement pertinent d'utiliser des modèles de substitution lorsque des éval-



uations de  $u(\cdot; y)$  pour de nombreuses valeurs de  $y$  sont demandées. La construction d'un métamodèle pour les fonctions en grande dimension est aujourd'hui un sujet de grand intérêt, par exemple en quantification d'incertitude (UQ) [97], pour des problèmes de contrôle ou d'optimisation.

De ce qui précède, au moins deux questions émergent, que nous aborderons dans ce manuscrit :

- Comment approcher la solution  $u(x)$  d'une EDP lorsque  $d$  est grand ?
- Comment approcher la variété des solutions  $\mathcal{M} := \{u(\cdot; y) : y \in Y\}$  par des sous-espaces de faible dimension lorsque  $p$  est grand ?

Pour aborder ces questions, nous supposons que la solution  $u$  admet une certaine représentation probabiliste, connue sous le nom de représentation de Feynman-Kac, qui trouve son origine dans les travaux de Richard Feynman et Mark Kac dans les années 1960 [65]. Cette représentation est au cœur des contributions de cette thèse.

## EDPs en grande dimension

Nous nous concentrons d'abord sur l'approximation de la solution  $u(x)$  d'une EDP en grande dimension. Nous sommes donc dans le cas où  $u : \mathcal{O} \subset \mathbb{R}^d \rightarrow \mathbb{R}$  est une fonction en grande dimension ( $d \gg 1$ ), avec  $\mathcal{O}$  un domaine physique (spatial, ou spatio-temporel).

Les méthodes numériques standards pour résoudre les EDPs comprennent notamment la méthode des éléments finis (FEM) [104], les différences finies [96] ou encore les volumes finis. Ces méthodes permettent d'approcher la solution globale  $u : \mathcal{O} \rightarrow \mathbb{R}$  de l'EDP. Elles sont très efficaces pour résoudre des problèmes en dimension faible (disons  $d \leq 3$ ). Cependant, elles nécessitent souvent l'utilisation d'une discrétisation fine du domaine physique  $\mathcal{O}$ , ce qui rend impossible leur utilisation pour les problèmes en grande dimension même pour les ordinateurs actuels. D'autres approches sans maillage ont été introduites plus récemment, comme [39]. Cependant, à notre connaissance, elles n'ont pas encore été adaptées pour des problèmes en grande dimension.

Pour lutter contre la malédiction de la dimension, des méthodes d'approximation par réseaux de tenseurs ou réseaux de neurones ont été proposées [4, 70, 56]. Des méthodes d'approximation parcimonieuses ont également été développées [19]. Ces techniques ont été exploitées pour la résolutions d'EDPs en grande dimension, voir par exemple [94, 95].

Pour en venir à des techniques approximation plus locales, par opposition aux techniques globales, nous rappelons qu'il est possible d'obtenir des évaluations ponctuelles de la solution en utilisant une représentation probabiliste de celle-ci. La représentation dite de Feynman-Kac mentionnée précédemment fait partie de ces représentations probabilistes. De plus, il n'y a pas de malédiction de la dimension à affronter

puisque nous nous concentrons uniquement sur l'obtention d'évaluations ponctuelles. Cette représentation existe pour plusieurs classes d'EDP linéaires et non linéaires [47]. Elle permet d'écrire la solution  $u$  sous la forme

$$(1) \quad u(x) = \mathbb{E}[Q(X^x)], \quad \forall x \in \mathcal{O},$$

où  $Q$  est une fonction dépendant des données de l'EDP et d'un certain processus de diffusion stochastique  $X^x$  dépendant de l'EDP et du point auquel la solution  $u$  est évaluée. En utilisant une estimation de Monte-Carlo traditionnelle basée sur  $M$  échantillons i.i.d. de  $X^x$ , nous avons accès à des évaluations ponctuelles bruitées de  $u$  en différents points  $x \in \mathcal{O}$ . Une analyse standard montre que la précision de ces évaluations ponctuelles se comporte généralement en  $O(\sqrt{M^{-1}} + \Delta t^\alpha)$  pour un certain  $\alpha > 0$ , où  $\Delta t$  est un paramètre d'intégration temporelle pour l'approximation du processus stochastique  $X^x$  [71]. Lorsque l'on souhaite approcher précisément  $u$ , par exemple jusqu'à la précision machine, les coûts de calcul deviennent dès lors trop élevés, même pour les ordinateurs actuels.

Différentes méthodes permettent de réduire la variance des évaluations ponctuelles de la solution, parmi lesquelles on peut citer [44, 76]. Les auteurs dans [48] ont introduit une méthode de réduction de variance par variables de contrôle globales en utilisant la représentation de Feynman-Kac. Dans [49] ils ont décrit et analysé plus précisément l'utilisation de ces variables de contrôle. Ils construisent d'abord une approximation globale  $\tilde{u}$  de  $u$  dans un espace d'approximation de dimension finie  $\mathcal{P}$ . Ils utilisent par exemple pour cela des méthodes d'interpolation reposant sur des estimations d'évaluations ponctuelles de  $u$  sur une grille unisolvante  $\Gamma_{\mathcal{P}}$  pour  $\mathcal{P}$ , obtenues par des méthodes de Monte-Carlo pour la représentation probabiliste (1). Montrant ensuite que l'erreur  $e = u - \tilde{u}$  admet également une représentation probabiliste, les auteurs construisent une approximation globale  $\tilde{e}$  de  $e$  basée sur des estimations d'évaluations ponctuelles obtenues par des méthodes de Monte-Carlo. Le processus peut ensuite être itéré plusieurs fois pour  $\tilde{u} + \tilde{e}$ . Il en résulte des évaluations ponctuelles approchées de  $u$  sur la grille  $\Gamma_{\mathcal{P}}$  avec une erreur qui dépend de l'erreur d'intégration temporelle de l'erreur d'approximation dans  $\mathcal{P}$  de la solution exacte  $u$  [49].

## Première Contribution

Lorsque  $d \gg 1$ , l'espace d'approximation  $\mathcal{P}$  dans lequel vivent  $\tilde{u}$  et  $\tilde{e}$ , et la grille associée  $\Gamma_{\mathcal{P}}$  doivent être bien choisis. Ici, nous combinons la méthode des variables de contrôle de [49] avec la stratégie d'interpolation parcimonieuse adaptative de [22] qui construit de manière adaptative l'espace d'approximation  $\mathcal{P}$  et en même temps la grille unisolvante associée  $\Gamma_{\mathcal{P}}$ . Nous proposons deux versions adaptatives de l'algorithme de [49]. Étant donné une approximation globale  $\tilde{u}_k$  de  $u$ , notre premier algorithme construit au pas  $k$  une interpolation adaptative parcimonieuse  $\tilde{e}_k$  de l'erreur  $u - \tilde{u}_k$  dans l'espace d'approximation  $\mathcal{P}_k$  (en utilisant sa grille d'interpolation associée  $\Gamma_{\mathcal{P}_k}$ ). Notre deuxième contribution est un algorithme d'interpolation adaptative perturbée basé sur [22], où à chaque étape  $k$  de l'algorithme, nous calculons une approximation  $\tilde{u}$  de  $u$  en utilisant l'algorithme utilisant des variables de contrôle de [49], plutôt que l'interpolation sur  $\mathcal{P}_k$  de la solution exacte. Ce travail a été publié dans [11].

## EDPs paramétrées et Optimisation Discrète

Nous nous concentrons désormais sur l'approximation de  $y \in Y \mapsto u(\cdot; y) \in \mathcal{H}$ , qui est la solution d'une certaine EDP paramétrée, où  $\mathcal{H}$  est un espace de Hilbert de fonctions définies sur le domaine  $\mathcal{O}$ . Les méthodes de réduction de modèle visent à réduire la complexité des modèles numériques complexes. Parmi ces méthodes, on retrouve les méthodes de base réduite qui sont très pertinentes pour certaines classes d'EDPs paramétrées et qui exploitent une bonne approximabilité de la variété des solutions  $\mathcal{M} := \{u(y) : y \in Y\}$  par des espaces de faible dimension [22, 24, 26].

Les méthodes de base réduite construisent des sous-espaces  $H_n$  de dimension  $n$  dans  $\mathcal{H}$  tels que la distance entre  $H_n$  et  $\mathcal{M}$  est proche de la  $n$ -largeur de Kolmogorov [90] qui caractérise la plus petite distance entre un sous-espace de dimension  $n$  de  $\mathcal{H}$  et  $\mathcal{M}$ , c'est-à-dire la meilleure erreur d'approximation qui peut être atteinte par les outils d'approximation linéaire. Nous nous concentrons en particulier sur les algorithmes gloutons pour les méthodes de base réduite [13, 18].

Dans un cadre idéalisé, les algorithmes gloutons construisent de manière adaptative une séquence imbriquée d'espaces réduits  $(H_n)_{n \geq 1}$  selon la procédure suivante. À une étape  $n \geq 1$ , l'espace réduit courant  $H_{n-1}$  est enrichi avec un snapshot  $u(y^n)$  qui est sélectionné parmi toutes les éléments de  $\mathcal{M}$  dont la distance à  $H_{n-1}$  est maximale. Cette distance est en pratique estimée à l'aide d'une estimation d'erreur a posteriori [92] peu coûteuse à calculer. Pour les EDPs paramétrées, une estimation d'erreur standard utilise la norme du résidu [54, 53]. De telles estimations d'erreur dégradent généralement la performance de l'algorithme glouton, qui se transforme alors en un algorithme glouton sous-optimal [32]. Pour certains problèmes paramétrés, les méthodes d'estimation d'erreur basées sur le résidu donnent de mauvais résultats dans la construction de l'espace réduit. Des techniques alternatives ont donc été proposées, comme le préconditionnement [103] ou l'estimation d'erreur sans résidu [20].

Cependant, pour des applications pratiques, la maximisation de l'erreur d'approximation sur un ensemble continu  $Y$  reste impossible. C'est pourquoi on considère généralement des ensembles d'apprentissage discrets  $\Xi \subset Y$  et la sélection des snapshots est faite parmi les éléments de la variété discrète des solutions  $\mathcal{M}_\Xi := \{u(\xi) : \xi \in \Xi\}$ . Le choix de ces ensembles d'apprentissage discrets est particulièrement difficile pour les espaces de paramètres de grande dimension, c'est-à-dire lorsque  $p \gg 1$ . La sélection aléatoire de ces ensembles d'apprentissage semble être une technique intéressante lorsque la variété des solutions  $\mathcal{M}$  peut être bien approchée par des espaces de faible dimension [27]. Il en résulte un algorithme glouton probabiliste.

La norme de l'erreur dans les algorithmes gloutons peut être écrite comme une espérance utilisant une représentation de Feynman-Kac de ladite erreur. Dans ce contexte, afin d'enrichir l'espace réduit avec un snapshot sélectionné à l'aide d'un algorithme glouton, nous devons résoudre un problème d'optimisation discrète du

type suivant

$$(2) \quad \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)],$$

où, dans le contexte des méthodes de base réduite, les  $(Z(\xi))_{\xi \in \Xi}$  sont des variables aléatoires dépendant de l'espace réduit. De tels problèmes d'optimisation discrète ont déjà été largement étudiés depuis les années 1950 [91, 73], initialement pour des applications médicales. Les algorithmes de bandit utilisent des estimations de Monte-Carlo et des inégalités de concentration afin de fournir une solution quasi-optimale de (2) avec une probabilité élevée. Le résultat de tels algorithmes est un paramètre  $\hat{\xi}$  dans  $\Xi$  qui est *probably approximately correct* (PAC) [38] dans un sens absolu, ce qui signifie que

$$\mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \epsilon \right) \geq 1 - \lambda,$$

pour des valeurs prescrites de  $\epsilon$  dans  $(0, 1)$  et  $\lambda$  dans  $(0, 1)$ , avec  $\xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)]$ .

## Seconde contribution

Les algorithmes de bandits existants ont besoin d'estimer intelligemment  $\epsilon$  afin de pouvoir être utilisés pour la maximization de l'erreur à posteriori dans des algorithmes gloutons. Pour contourner l'estimation de  $\epsilon$ , la deuxième contribution de cette thèse est un algorithme probablement approximativement correct (PAC) en précision relative qui combine des estimations de Monte-Carlo avec une précision relative garantie [79] avec un algorithme PAC [38]. Cet algorithme fournit  $\hat{\xi}$  dans  $\Xi$  qui satisfait

$$(3) \quad \mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \epsilon |\mathbb{E}[Z(\xi^*)]| \right) \geq 1 - \lambda,$$

pour des valeurs prescrites de  $\epsilon$  dans  $(0, 1)$  et  $\lambda$  dans  $(0, 1)$ , avec  $\xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)]$ .

## Troisième contribution

La troisième contribution de cette thèse combine les méthodes de base réduite) avec une sélection de snapshots basée sur le nouvel algorithme PAC en précision relative. En réinterprétant l'erreur d'approximation dans un algorithme glouton comme une espérance, nous pouvons sélectionner un snapshot pour enrichir la base réduite en utilisant l'algorithme PAC en précision relative. Il en résulte un algorithme glouton probabiliste. En combinant cet algorithme avec l'utilisation d'ensembles d'apprentissage aléatoires [27], qui sont adaptés au cas  $p \gg 1$ , nous proposons un deuxième algorithme glouton probabiliste qui consiste à choisir aléatoirement un snapshot quasi-optimale parmi des ensembles d'apprentissage aléatoires en utilisant l'algorithme PAC en précision relative que nous avons proposé.

## Plan du Manuscrit

Cette thèse est composée de cinq chapitres.

Dans un premier temps, nous nous concentrons sur la résolution d'EDPs en grande dimension à l'aide de méthodes probabilistes, ce qui correspond à notre première contribution, publiée dans [11]. Un aperçu des méthodes probabilistes pour les EDPs linéaires est présenté dans le Chapitre 1 tandis que notre contribution est présentée dans le Chapitre 2.

Le Chapitre 3 traite d'un problème d'optimisation discrète pour une fonction définie comme une espérance, et introduit notre deuxième contribution. Ce travail a été soumis comme [12].

Enfin, nous étudions les EDPs paramétrées. Dans le Chapitre 4, nous donnons un aperçu des méthodes de base réduite pour ces modèles. Le Chapitre 5 introduit deux algorithmes gloutons probabilistes pour les méthode de base réduite, qui utilisent l'algorithme d'optimisation discrète introduit dans le Chapitre 3.

# Introduction (english version)

This thesis is devoted to the approximation of the solution of high-dimensional partial differential equations (PDEs), possibly parameter-dependent. Approximating the solution of PDEs is a vast and quite old field of mathematics: PDEs have been analysed since the late 17<sup>th</sup> century by Newton and Leibniz. While at this time only simple models with analytical solutions were studied, the recent developments in computer science have opened up the possibility to address more complicated PDEs through numerical methods. More recently parameter-dependent PDEs have been considered to study more complex problems in optimization, control, sensitivity analysis or uncertainty quantification.

The solution of a PDE is formally defined as a multi-variable function  $u(x)$  depending on some physical (spatial, temporal) variables  $x$  in  $\mathbb{R}^d$ . For parameter-dependent PDEs, the solution  $u(x; y)$  depends on additional variables  $y \in Y \subset \mathbb{R}^p$ .

The concrete problems for which PDEs are used arise in various fields, for example in physics, biology or finance. These applications usually require to be able to approximate up to some precision the solution  $u$  of a PDE. For parameter-dependent PDEs, precise approximations of  $u(\cdot; y)$  for many instances  $y$  of the parameters are usually required. Precise approximations usually means fine, thus costly, numerical models.

Moreover, the numerical complexity to approximate functions  $u(x)$  or  $u(x; y)$  might increase exponentially with the dimensions  $d$  or  $p$ . This is known as the curse of dimensionality [88, 23]. This curse can sometimes be avoided, for example using Monte-Carlo methods for the estimation of integrals [85]. When it comes to function approximation, which is a more complex problem than integration, avoiding this curse is not always possible using random methods [72].

High-dimensional PDEs thus result in difficult approximation problems, in both parameter-independent and parameter-dependent cases. For parameter-dependent PDEs, a solution is to use a metamodel, also known as surrogate model, which approximates  $u$  and is cheap to evaluate compared to the initial numerical model. It is particularly relevant to use surrogates when evaluations of  $u(\cdot; y)$  for many values of  $y$  are required. The building of metamodel for high-dimensional functions is nowadays a matter of high interest, e.g. in uncertainty quantification (UQ) [97], control

or optimisation.

From the foregoing, at least two questions arise, that we will address in this manuscript:

- How to approximate the solution  $x \in \mathbb{R}^d \mapsto u(x)$  of a PDE when  $d$  is large ?
- How to approximate the solution manifold  $\mathcal{M} := \{u(\cdot; y) : y \in Y \subset \mathbb{R}^p\}$  by low-dimensional subspaces of  $\mathcal{H}$  when  $p$  is large ?

To tackle these questions, we will assume that the solution  $u$  admits some probabilistic representation, known as the Feynman-Kac representation, that originates in the work from Richard Feynman and Mark Kac in the 1960s [65]. This representation is at the core of the contributions of this thesis.

## High-Dimensional PDEs

First we focus on the approximation of the solution  $u(x)$  to some high-dimensional PDE. We are thus in the case where  $u : \mathcal{O} \subset \mathbb{R}^d \rightarrow \mathbb{R}$  is a high-dimensional function ( $d \gg 1$ ), with  $\mathcal{O}$  a physical domain (spatial or spatio-temporal).

Standard numerical methods for solving PDEs include Finite Element Method (FEM) [104], finite differences [96] or finite volumes. Such methods approximate the global solution  $u : \mathcal{O} \rightarrow \mathbb{R}$ . They are really efficient for solving low-dimensional problems (say  $d \leq 3$ ). However they often require the use of a fine discretization of the physical domain  $\mathcal{O}$ , which makes them intractable for high-dimensional problems, even for today's computers. Other meshless approaches have been introduced more recently, such as [39]. However they still have not been considered, up to our knowledge, for high-dimensional problems.

To tackle the curse of dimensionality, methods based on tensor networks or neural networks have been proposed [4, 70, 56]. Sparse approximation methods have also been developed [19] and exploited to solve high-dimensional PDEs, see e.g. [94, 95].

Moving to local approximation methods, we recall that it is possible to get pointwise evaluations of the solution using a probabilistic representation. The so-called Feynman-Kac representation mentioned earlier is such a probabilistic representation. There is no curse of dimensionality to tackle since we only focus on getting a few pointwise evaluations. This representation exists for several classes of linear and non-linear PDEs [47] and yields an expression of the solution of the form

$$(4) \quad u(x) = \mathbb{E}[Q(X^x)], \quad \forall x \in \mathcal{O},$$

where  $Q$  is a functional depending on the data of the PDE and some stochastic diffusion process  $X^x$  depending on the PDE and on the point at which the solution  $u$  is evaluated. Using a traditional Monte-Carlo estimate based on  $M$  i.i.d. samples of  $X^x$ , we have access to noisy pointwise evaluations of  $u$  in different points  $x \in \mathcal{O}$ .

A standard analysis shows that the precision of these pointwise evaluations generally behaves as  $O(\sqrt{M^{-1}} + \Delta t^\alpha)$  for some  $\alpha > 0$ , where  $\Delta t$  is a time-integration parameter for the approximation of the stochastic process [71]. When we wish to approximate precisely  $u$ , e.g. up to machine precision, the computational costs are too high, even for today's computers.

Different methods allow to reduce the variance of the pointwise evaluations of the solution, among them we mention [44, 76]. The authors in [48] introduced a variance reduction method with global control variates using the Feynman-Kac representation. In [49], they described and analysed more precisely the use of these control variates. First they build a global approximation  $\tilde{u}$  of  $u$  in some finite-dimensional approximation space  $\mathcal{P}$ . They use for example interpolation methods relying on estimations of pointwise evaluations of  $u$  on an unisolvent grid  $\Gamma_{\mathcal{P}}$  for  $\mathcal{P}$ , obtained by Monte-Carlo methods for the probabilistic representation (4). Showing then that the error  $e = u - \tilde{u}$  admits a probabilistic representation, the authors build a global approximation  $\tilde{e}$  of  $e$  based on estimations of pointwise evaluations obtained by Monte-Carlo methods. Updating  $\tilde{u}$  to  $\tilde{u} + \tilde{e}$ , the process can be then iterated multiple times. It results in approximate pointwise evaluations of  $u$  on the grid  $\Gamma_{\mathcal{P}}$  with an error depending on the time integration error of the approximation error in  $\mathcal{P}$  of the exact solution  $u$  [49].

## First contribution

When  $d \gg 1$ , the approximation space  $\mathcal{P}$  in which lives  $\tilde{u}$  and  $\tilde{e}$ , and the associated grid  $\Gamma_{\mathcal{P}}$  must be well-chosen. Here we combine the control variate method from [49] with the adaptive interpolation strategy from [22] which builds adaptively and at the same time the approximation space  $\mathcal{P}$  and the adapted unisolvent grid  $\Gamma_{\mathcal{P}}$ . We propose two adaptive versions of the algorithm of [49]. Being given a global approximation  $\tilde{u}_k$  of  $u$ , our first algorithm builds at step  $k$  a sparse adaptive interpolation  $\tilde{e}_k$  of the error  $u - \tilde{u}_k$  in the approximation space  $\mathcal{P}_k$  (and its associated grid  $\Gamma_{\mathcal{P}_k}$ ). Our second contribution is a perturbed adaptive interpolation algorithm based on [22], where at each step  $k$  of the algorithm, we compute an approximation  $\tilde{u}$  of  $u$  using the control variate algorithm from [49], rather than the exact interpolation on  $\mathcal{P}_k$ . This work can be found in [11].

## Parameter-Dependent PDEs and Discrete Optimization

We focus on approximating the map  $y \in Y \mapsto u(\cdot; y) \in \mathcal{H}$ , which is the solution of some parameter-dependent PDE, with  $\mathcal{H}$  a Hilbert space of functions defined on  $\mathcal{O}$ . Model order reduction (MOR) methods aim at reducing the complexity of complex numerical models. Among MOR we can find reduced basis methods (RBM) that are highly relevant for some classes of parameter-dependent PDEs and exploit a good approximability of the solution manifold  $\mathcal{M} := \{u(y) : y \in Y\}$  by low-dimensional



spaces [22, 24, 26].

RBM build  $n$ -dimensional subspaces  $H_n$  of  $\mathcal{H}$  such that the distance between  $H_n$  and  $\mathcal{M}$  is close to the Kolmogorov  $n$ -width [90] that characterizes the smallest distance between a  $n$ -dimensional subspace of  $\mathcal{H}$  and  $\mathcal{M}$ , that is the best approximation error that can be achieved by linear approximation tools. We focus particularly on greedy algorithms for RBM [13, 18].

In an idealized framework, greedy algorithms adaptively build a nested sequence of reduced space  $(H_n)_{n \geq 1}$  according to the following procedure. At each step  $n \geq 1$ , the current reduced space  $H_{n-1}$  is enriched with a snapshot  $u(y^n)$  which is selected among all the elements from  $\mathcal{M}$  whose distance to  $H_{n-1}$  is maximal. This distance is in practice estimated using an a posteriori error estimate [92] inexpensive to compute. For parameter-dependent PDEs, a traditional error estimate is based on the norm of the residual [54, 53]. Such error estimates usually deteriorate the performance of the greedy algorithm, that turns into a weak-greedy algorithm [32]. For some parameter-dependent problems (with low decay of the Kolmogorov  $n$ -width of their solution manifold) residual-based error estimates may yield a poor quality of the reduced space. Thus alternative techniques have been proposed, such as preconditioning [103] or residual-free error estimate [20].

However, in practical applications, maximizing the approximation error over a continuous set  $Y$  remains impossible. This is why discrete training sets  $\Xi \subset Y$  are usually considered and the selection of the snapshots is made among elements of the discretized solution manifold  $\mathcal{M}_\Xi := \{u(\xi) : \xi \in \Xi\}$ . The choice of these discrete training sets is particularly challenging for high-dimensional parameter spaces, i.e. when  $p \gg 1$ . A random selection of these training sets seems an interesting technique when the solution manifold  $\mathcal{M}$  can be well-approximated by low dimensional spaces [27]. This results in a probabilistic greedy algorithm.

The norm of the a posteriori error estimate in greedy algorithms can be written as an expectation using a Feynman-Kac representation of the error. In this context, in order to enrich the reduced space with a snapshot selected using a greedy procedure, we need to solve a discrete optimization problem of the following type

$$(5) \quad \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)],$$

where, in the context of reduced basis methods, the  $(Z(\xi))_{\xi \in \Xi}$  are random variables depending on the reduced space. Such discrete optimization problems have already been widely studied since the 1950s [91, 73] and were originally mentioned in medical applications. Bandit algorithms use Monte-Carlo estimates and concentration inequalities in order to provide a quasi-optimal solution of (5) with high probability. The result of such algorithms is a parameter  $\hat{\xi}$  in  $\Xi$  that is probably approximately correct (PAC) [38] in an absolute sense, which means

$$\mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \epsilon \right) \geq 1 - \lambda,$$

for some prescribed values of  $\epsilon$  in  $(0, 1)$  and  $\lambda$  in  $(0, 1)$ , with  $\xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)]$ .

## Second contribution

Existing bandit algorithms would need a careful estimation of  $\epsilon$  in order to preserve guarantees of weak-greedy algorithms. To circumvent the estimation of  $\epsilon$ , the second contribution of this thesis is a probably approximately correct (PAC) algorithm in relative precision which combines Monte-Carlo estimates with guaranteed relative precision [79] with probably approximately correct (PAC) algorithm [38]. It returns  $\hat{\xi}$  in  $\Xi$  which satisfies

$$(6) \quad \mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \epsilon |\mathbb{E}[Z(\xi^*)]| \right) \geq 1 - \lambda,$$

for some prescribed values of  $\epsilon$  in  $(0, 1)$  and  $\lambda$  in  $(0, 1)$ , with  $\xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)]$ .

## Third contribution

The third contribution of this thesis combines reduced basis methods (RBM) with a snapshot selection based on the new PAC algorithm in relative precision. By reinterpreting the approximation error in a greedy algorithm as an expectation, we can select a snapshot to enrich the reduced basis using the PAC algorithm in relative precision. This results in a probabilistic greedy algorithm. Combining this algorithm with the use of random training sets [27], that are adapted to the case  $p \gg 1$ , we propose a second probabilistic greedy algorithm that consists in randomly choosing a quasi-optimal snapshot among random training sets using the PAC algorithm in relative precision we proposed.

## Thesis' Outline

This thesis is composed of five chapters.

First we focus on solving high-dimensional PDEs using probabilistic methods, which corresponds to our first contribution, published in [11]. A survey of probabilistic methods for linear parameter-independent PDEs is presented in Chapter 1 while the contribution is presented in Chapter 2.

In Chapter 3 we consider a discrete optimization problem for a function defined as an expectation, and introduces to our second contribution. This work has been submitted as [12].

Finally we study parameter-dependent PDEs. In Chapter 4, we survey reduced basis methods for parameter-dependent PDEs. Chapter 5 introduces two probabilistic greedy algorithms for reduced basis methods based on the discrete optimization algorithm from Chapter 3.



# Chapter 1

## Probabilistic Methods for PDEs: a Survey

This chapter first introduces probabilistic representation for parameter-independent partial differential equations (PDEs) on bounded domains. Then numerical methods to simulate pointwise evaluations of the solution of parameter-independent PDEs using their probabilistic representation are surveyed.

### Contents

1.1	Stochastic Differential Equations . . . . .	29
1.2	Feynman-Kac Representation . . . . .	31
1.2.1	Backward Parabolic Boundary Value Problem . . .	32
1.2.2	Elliptic Boundary Value Problem . . . . .	33
1.3	Numerical Methods . . . . .	34
1.3.1	Approximation of Diffusion Processes . . . . .	35
1.3.2	Monte-Carlo Methods . . . . .	35
1.3.3	Statistical Error for the Simulation of Pointwise Evaluations . . . . .	36

We consider either elliptic or parabolic partial differential equations (PDEs) defined on a physical domain  $\mathcal{O} \subset \mathbb{R}^d$ . This physical domain is an open bounded spatial domain, denoted by  $\mathcal{D} \subset \mathbb{R}^d$  with a boundary  $\partial\mathcal{D}$ , in the elliptic case. In the parabolic case  $\mathcal{O}$  corresponds to a spatio-temporal domain, typically  $[0, T] \times \mathcal{D} \subset \mathbb{R} \times \mathbb{R}^d$ . The space variable is denoted by  $x = (x_1, \dots, x_d)$ . In the parabolic case, the time variable is denoted by  $t$ . We denote by  $\mathcal{L}$  a linear differential operator taken under the non-divergence form

$$(1.1) \quad \begin{aligned} \mathcal{L}(u)(t, x) = & \frac{1}{2} \sum_{i,j=1}^d (\sigma(t, x)\sigma(t, x)^T)_{ij} \partial_{x_i x_j}^2 u(t, x) \\ & + \sum_{i=1}^d b_i(t, x) \partial_{x_i} u(t, x), \end{aligned}$$

where  $b = (b_1, \dots, b_d) : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}^d$  and  $\sigma : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}^{d \times d}$ .

With these notations, the backward parabolic boundary-value problem is composed of the following equations satisfied by the unknown solution  $u : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$

$$(1.2) \quad \partial_t u + \mathcal{L}(u) - ku + g = 0 \quad \text{in} \quad [0, T] \times \mathcal{D},$$

$$(1.3) \quad u(T, x) = f(T, x) \quad \text{for} \quad x \in \overline{\mathcal{D}},$$

$$(1.4) \quad u(t, x) = f(t, x) \quad \text{for} \quad (t, x) \in [0, T] \times \partial\mathcal{D}.$$

Equation (1.2) is a parabolic PDE. We have  $k : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$ , and  $f : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$  represents both the terminal and boundary conditions (respectively (1.3) and (1.4)). The function  $g : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$  is the source term of the PDE. In the literature, the set of equations (1.2)-(1.4) is also referred to as the Cauchy-Dirichlet problem.

We study also elliptic problems, where the time variable does not appear. It is composed of the following equations satisfied by the unknown solution  $u : \overline{\mathcal{D}} \rightarrow \mathbb{R}$

$$(1.5) \quad -\mathcal{L}(u) + ku = g \quad \text{in} \quad \mathcal{D},$$

$$(1.6) \quad u = f \quad \text{on} \quad \partial\mathcal{D}.$$

Equation (1.6) stands for a Cauchy boundary condition. The functions  $k : \overline{\mathcal{D}} \rightarrow \mathbb{R}$ ,  $f : \partial\mathcal{D} \rightarrow \mathbb{R}$  and  $g : \overline{\mathcal{D}} \rightarrow \mathbb{R}$  are not time dependent in this case. Neither are the functions  $b : \overline{\mathcal{D}} \rightarrow \mathbb{R}^d$  and  $\sigma : \overline{\mathcal{D}} \rightarrow \mathbb{R}^{d \times d}$  defining the coefficients of the linear differential operator (1.1).

**Remark 1.1.** *The backward formulation in the parabolic case arises from problems in finance. In physics, forward problems are more common, where the terminal condition (1.3) is replaced by an initial condition. The references in the literature on theoretical results for PDEs usually treat the forward parabolic case. However, using the change of variable  $t \rightarrow T - t$ , it is easy to derive theoretical results for the backward case. Moreover, both elliptic and parabolic problems can also be considered on unbounded domains. In order to simplify our study we will not consider PDEs on unbounded domains here and we refer the reader to the following non-exhaustive literature [28, 41, 68, 47] for more details.*

No assumption were given yet on the different functions involved in the definition of the problems (1.2)-(1.4) and (1.5)-(1.6). For both parabolic and elliptic problems, uniqueness of the solution can be obtained using the so-called maximum-principle. We refer to [36, Chapter 2.3, Theorem 5] for the parabolic problem and to [36, Chapter 2.2, Theorem 5] for the elliptic problem. In these results, the uniqueness relates to classical solutions that are solutions in  $\mathcal{C}([0, T] \times \overline{\mathcal{D}}) \cap \mathcal{C}^{1,2}([0, T] \times \mathcal{D})$ , which means continuous over its domain of definition with continuous derivatives. It ensures that equation (1.2) is well-defined. Existence of classical solutions can be obtained too and we refer to [41, Chapter 6, Theorem 3.6] and references therein for the forward parabolic problem and to [41, Chapter 6, Theorem 2.4] and references therein for the elliptic problem.

**Remark 1.2.** *A weaker notion of solution exists. It is related to the weak formulation of PDEs. We mention this aspect in Section 4.3.1*

This chapter is a survey of probabilistic methods for both elliptic and parabolic problems defined above. We recall that such methods give access to pointwise evaluations of the solution  $u$ , i.e. to  $u(t, x)$  for chosen values of  $(t, x)$  in  $[0, T] \times \overline{\mathcal{D}}$  in the parabolic case, and to  $u(x)$  for chosen values of  $x$  in  $\overline{\mathcal{D}}$  in the elliptic case. Stochastic Differential Equations (SDEs) are at the core of the probabilistic methods introduced hereafter. We first introduce in Section 1.1 the notion of SDE. Then we show in Section 1.2 how to express the solution of a PDE as the expectation, depending on a diffusion process solution of a well-chosen SDE, related to data of the problem. This being done, we present numerical methods allowing to estimate pointwise evaluations of the solution  $u$ . Since the numerical methods are similar for both elliptic and parabolic problems, we present them in the parabolic case in Section 1.3 and adapt this presentation for the elliptic case in Section 2.2.

## 1.1 Stochastic Differential Equations

Stochastic calculus originates in the description of Brownian motion in the early 19<sup>th</sup> century by Robert Brown, which characterizes a particle with erratic movements. These observations have led to the introduction of the notions of stochastic process and stochastic integrals that are at the core of the Itô calculus, introduced in the 1940s. Such notions give a rigorous framework for the definition of an SDE. We recall the most important ones here and refer the reader to [46] and references therein for more details. First, we introduce a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ , where  $\Omega$  is a sample space,  $\mathcal{F}$  a  $\sigma$ -algebra on  $\Omega$  and  $\mathbb{P}$  a probability function on  $\mathcal{F}$ .

**Definition 1.3** (Stochastic process with values in  $\mathbb{R}^d$ ). *Let  $\mathbb{T}$  be a subset of  $\mathbb{R}$ . A stochastic process indexed by  $\mathbb{T}$  is a family  $(X_t)_{t \in \mathbb{T}}$  of random variables defined on the probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ . The random variables  $X_t$  may be valued in any measurable space  $(\mathbb{R}^d, \mathcal{G})$ , where  $\mathcal{G}$  is a  $\sigma$ -algebra on  $\mathbb{R}^d$ .*

Then we define a measurable (stochastic) process.

**Definition 1.4** (Measurable and adapted process). *A real-valued stochastic process  $(X_t)_{t \in \mathbb{T}}$  is measurable if  $X : (t, \omega) \in (\mathbb{T} \times \Omega, \mathcal{B}(\mathbb{T}) \otimes \mathcal{F}) \rightarrow X_t(\omega) \in (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$  is  $(\mathcal{B}(\mathbb{T}) \otimes \mathcal{F}, \mathcal{B}(\mathbb{R}^d))$ -measurable.*

*Recalling that a filtration is a family  $(\mathcal{F}_t)_{t \in \mathbb{T}} \subset \mathcal{F}$  such that*

$$t_1 \leq t_2 \Rightarrow \mathcal{F}_{t_1} \subset \mathcal{F}_{t_2},$$

*$(X_t)_{t \in \mathbb{T}}$  is said to be adapted to the filtration  $(\mathcal{F}_t)_{t \in \mathbb{T}}$  if, for all  $t \geq 0$ ,  $X_t : (\Omega, \mathcal{F}_t) \rightarrow (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$  is  $(\mathcal{F}_t, \mathcal{B}(\mathbb{R}^d))$ -measurable.*

Now we give a definition of the (standard) Brownian motion in dimension one, which is a particular stochastic process indexed by  $\mathbb{T} = \mathbb{R}^+$ .

**Definition 1.5** (One-dimensional Brownian motion). *A (standard one-dimensional) Brownian motion is a real-valued stochastic process  $(W_t)_{t \in \mathbb{R}^+}$ , with continuous trajectories, such that*

1.  $W_0 = 0$  ;
2. any time increment  $W_t - W_s$  ( $0 \leq s < t$ ) has the distribution  $\mathcal{N}(0, t - s)$  ;
3. for any  $0 = t_0 < t_1 < \dots < t_n$ , the increments  $\{W_{t_{i+1}} - W_{t_i}; 0 \leq i \leq n - 1\}$  are independent.

From this definition, we can deduce the main properties of the Brownian motion. It includes the symmetry property (if  $W$  is a Brownian motion, so is  $-W$ ). We refer the reader to [47, Chapter IV.1] for additional properties.

The extension to obtain a  $d$ -dimensional Brownian motion consists simply in considering  $W = (W_1, \dots, W_d)$ , where each  $W_i$  is a one-dimensional Brownian motion, each one being independent from the others. In what follows, we will denote by  $(\mathcal{F}_t)_{t \in \mathbb{R}^+}$  the natural filtration of the Brownian motion. Unless otherwise specified, any adapted process will now be meant adapted to  $(\mathcal{F}_t)_{t \in \mathbb{R}^+}$ . We can then introduce the set of adapted and square-integrable processes indexed by  $\mathbb{T} = [0, T]$ , which is defined by

$$\mathbb{H}_T^2 := \mathbb{H}_{[0, T]}^2 = \{\phi \text{ adapted to } (\mathcal{F}_t)_{t \in [0, T]} \text{ such that } \|\phi\|_{\mathbb{H}_T^2}^2 := \mathbb{E} \left[ \int_0^T |\phi_t|^2 dt \right] < \infty\}.$$

We also define the set of elementary processes,  $\mathbb{H}^{\text{elem}}$ , which contains the processes  $(\phi_t)_{0 \leq t \leq T}$ , such that  $\phi_t = \phi_{t_i}$  for all  $t_i < t \leq t_{i+1}$ , where  $t_1 < t_2 < \dots$  is any time grid. We now introduce the notion of Stochastic Differential Equation (SDE) and conditions for existence and uniqueness of a solution to this SDE.

**Theorem 1.6** (Existence and uniqueness of a solution to a SDE). *Let  $W$  be a  $d$ -dimensional Brownian motion. Take  $s \in \mathbb{R}^+$  and  $s < S < +\infty$ . Let  $b : [s, S] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  and  $\sigma : [s, S] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$  be two continuous functions satisfying the following conditions for a constant  $C_{b, \sigma}$*

$$(A1) \quad \forall (t, x, y) \in [s, S] \times \mathbb{R}^d \times \mathbb{R}^d \\ |b(t, x) - b(t, y)| + |\sigma(t, x) - \sigma(t, y)| \leq C_{b, \sigma} |x - y|;$$

$$(A2) \quad \sup_{s \leq t \leq S} |b(t, 0)| + |\sigma(t, 0)| \leq C_{b, \sigma}.$$

For a given  $x_s \in \mathbb{R}^d$  and for any  $s \leq t \leq S$  we consider the following SDE

$$X_t = x_s + \int_s^t b(r, X_r) dr + \int_s^t \sigma(r, X_r) dW_r. \quad (1.7)$$

There exists a unique process  $(X_t)_{t \in [s, S]}$  satisfying eq. (1.7), adapted to  $(\mathcal{F}_t)_{t \in [s, S]}$  and in  $\mathbb{H}_{[s, S]}^2$ , which moreover satisfies  $\mathbb{E}(\sup_{s \leq t \leq S} |X_t|^2) < C(1 + |x_s|^2)$  for some constant  $C$  depending on  $S - s$  and  $C_{b, \sigma}$ .

The last integral in (1.7) is defined in the sense of Ito. In particular for processes  $\phi$  in  $\mathbb{H}_T^2 \cap \mathbb{H}^{\text{elem}}$  we have

$$(1.8) \quad \int_0^T \phi_r dW_r = \sum_{t_i \leq T} \phi_{t_i} (W_{T \wedge t_{i+1}} - W_{t_i}).$$

This definition can be extended for processes in  $\mathbb{H}_T^2$ . We refer the reader to [47, Chapter IV.2] for this extension and we recommend [68] for a complete presentation on stochastic integration.

## 1.2 Feynman-Kac Representation

Our goal here is to present the links between SDEs and PDEs. To do so, we introduce the notion of Ito process and the Ito formula.

**Definition 1.7** (Ito process). *Let  $(b_t)_{t \geq 0} : \mathbb{R}^+ \rightarrow \mathbb{R}^d$  and  $(\sigma_t)_{t \geq 0} = ((\sigma_{1,t}, \dots, \sigma_{d,t}))_{t \geq 0} : \mathbb{R}^+ \rightarrow \mathbb{R}^{d \times d}$  be two adapted processes satisfying  $\int_0^s (|b_r| + |\sigma_r|)^2 dr < +\infty$  almost surely for all  $s \geq 0$ . The Ito process  $(X_t)_{t \geq 0}$ , starting at  $x_0$ , with drift coefficient  $b$  and diffusion coefficient  $\sigma$  is the  $d$ -dimensional stochastic process defined by*

$$(1.9) \quad X_t = x_0 + \int_0^t b_r dr + \int_0^t \sigma_r dW_r.$$

**Remark 1.8.** *The process  $(\sigma_t)_{t \geq 0}$  can be taken more generally with values in  $\mathbb{R}^{d \times q}$ , for some  $q \in \mathbb{N}$  possibly different from  $d$ , the space dimension. We focus in this manuscript on the case  $q = d$ . Moreover, the definition of the Ito process has been taken using  $s = 0$  as a time reference even if it is still possible to make it start at any time  $s$ , just like the stochastic process solution of (1.7). However, we stick to this time origin to avoid introducing too much notation.*

**Theorem 1.9** (Ito formula). *Let  $G$  be a real-valued function in  $\mathcal{C}^{1,2}(\mathbb{R}^+ \times \mathbb{R}^d)$  and  $(X_t)_{t \geq 0}$  the  $d$ -dimensional Ito process defined by (1.9). Thus the process  $(Y_t = G(t, X_t))_{t \geq 0}$  is a 1-dimensional Ito process given by*

$$(1.10) \quad \begin{aligned} G(t, X_t) = & G(0, x_0) + \int_0^t \partial_t G(s, X_s) ds + \int_0^t \nabla_x G(s, X_s) b_s ds \\ & + \int_0^t \nabla_x G(s, X_s) \sigma_s dW_s + \frac{1}{2} \int_0^t \sum_{k,l=1}^d \partial_{x_k x_l}^2 G(s, X_s) [\sigma_s \sigma_s^T]_{kl} ds. \end{aligned}$$

Ito formula allows to introduce probabilistic representations for PDEs, also known as Feynman-Kac representations. It originates in the 1940s, with the PhD work of Richard Feynman, then developed by Mark Kac, in particular in [65]. Such a representation is obtained by seeing the differential operator (1.1) as an infinitesimal generator of the  $d$ -dimensional diffusion process  $X^{t,x}$  solution of the following SDE

$$(1.11) \quad X_s^{t,x} = X_t^{t,x} + \int_t^s b(r, X_r^{t,x}) dr + \int_t^s \sigma(r, X_r^{t,x}) dW_r, \quad X_t^{t,x} = x \in \overline{\mathcal{D}},$$

whose coefficients are defined using the coefficients of  $\mathcal{L}$ . Here  $W$  is a  $d$ -dimensional Brownian motion. The stochastic process  $X^{t,x}$  is uniquely defined, under suitable assumptions such as (A1) and (A2), by its value  $x$  at time  $t$  and the functions  $b$  and  $\sigma$ .



**Remark 1.10.** *In the elliptic case, the different functions involved are not time-dependent and we take the time origin at  $t = 0$  in (1.11). When there is no ambiguity, we simplify our notation with the convention  $X^{0,x} = X^x$ .*

Applying now the Ito formula to a well-chosen function of  $X^{t,x}$ , it is possible to obtain the Feynman-Kac representation of the solution of the considered problem (either (1.2)-(1.4) or (1.5)-(1.6)).

### 1.2.1 Backward Parabolic Boundary Value Problem

The following Feynman-Kac representation comes from [47, Theorem IV.4.5].

**Theorem 1.11** (Feynman-Kac formula for backward parabolic PDE on bounded domain). *Set  $T > 0$  and assume that*

(P1) *assumptions (A1) and (A2) are satisfied,*

(P2)  *$\mathcal{D} \subset \mathbb{R}^d$  is a bounded domain and we define  $\tau^{t,x} = \inf\{s > t : X_s^{t,x} \notin \mathcal{D}\}$  as the first exit time of  $X^{t,x}$  from  $\mathcal{D}$ ,*

(P3) *the boundary  $\partial\mathcal{D}$  is regular for  $X^{t,x}$  in the sense that*

$$(1.12) \quad \forall t \in [0, T], \forall x \in \partial\mathcal{D}, \mathbb{P}(\tau^{t,x} = 0) = 1,$$

(P4) *the functions  $f, g, k : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$  are continuous,*

(P5) *there exists a continuous function  $u : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$ , of class  $\mathcal{C}^{1,2}$  on all open subset of  $[0, T[ \times \mathcal{D}$  verifying (1.2)-(1.4).*

*Then  $u$  admits the following probabilistic representation*

$$(1.13) \quad u(t, x) = \mathbb{E}[G(u, X^{t,x})],$$

*where, setting  $\tau_T^{t,x} = \tau^{t,x} \wedge T$ , we have*

$$\begin{aligned} G(u, X^{t,x}) = & f(\tau_T^{t,x}, X_{\tau_T^{t,x}}^{t,x}) \exp\left(-\int_t^{\tau_T^{t,x}} k(r, X_r^{t,x}) dr\right) \\ & + \int_t^{\tau_T^{t,x}} g(s, X_s^{t,x}) \exp\left(-\int_t^s k(r, X_r^{t,x}) dr\right) ds. \end{aligned}$$

Assumption (P5) could be replaced by suitable assumptions on the coefficients of the PDE and the linear operator  $\mathcal{L}$ , see e.g. [41, Chapter 6, Theorem 3.6], that ensure the existence of a classical solution to (1.2)-(1.4). An essential regularity assumption is needed on the boundary. In some Feynman-Kac representations (see e.g. [41, Chapter 6, Theorem 5.2]), a high regularity is required for  $\partial\mathcal{D}$  in order to get rid of assumption (P5). However, in many practical examples (e.g. on polygonal domains) this regularity for the boundary is not verified. In order to avoid these assumptions, it is possible to refer to existence and uniqueness theorem [41, Chapter 6, Theorem 3.6]. The only needed assumption on the boundary  $\partial\mathcal{D}$  in this theorem is the existence of a barrier at every point of  $(0, T) \times \partial\mathcal{D}$ . This condition ensures

a certain regularity of the boundary and thus the existence of a classical solution. Other conditions exist, such as the exterior cone condition, also called Zaremba's cone condition, see [68, Chapter 4.2.C] for a discussion on regularity conditions of  $\partial\mathcal{D}$ . Other works have been conducted linking these conditions [77], but we will not develop them further in this manuscript.

### 1.2.2 Elliptic Boundary Value Problem

For the elliptic problem, we recall the following notation  $X^x = X^{0,x}$  (see Remark 1.10). The following result is taken from [28, Theorem 2.4].

**Theorem 1.12** (Feynman-Kac formula for elliptic PDE on bounded domain). *Assume that*

(E1)  $\mathcal{D}$  is an open connected bounded domain of  $\mathbb{R}^d$ , regular in the sense that, if  $\tau^x = \inf \{s > 0 : X_s^x \notin \mathcal{D}\}$  is the first exit time of  $\mathcal{D}$  for the process  $X^x$ , solution of (1.11), we have

$$\mathbb{P}(\tau^x = 0) = 1, \quad x \in \partial\mathcal{D},$$

(E2)  $b, \sigma$  are Lipschitz functions over  $\overline{\mathcal{D}}$ ,

(E3)  $f$  is continuous on  $\partial\mathcal{D}$ ,  $g$  and  $k \geq 0$  are Hölder-continuous functions on  $\overline{\mathcal{D}}$ ,

(E4) (uniform ellipticity assumption) there exists  $c > 0$  such that

$$\sum_{i,j=1}^d \left( \sigma(x) \sigma(x)^T \right)_{ij} \Upsilon_i \Upsilon_j \geq c \sum_{i=1}^d \Upsilon_i^2, \quad \Upsilon \in \mathbb{R}^d, \quad x \in \overline{\mathcal{D}}.$$

Then, there exists a unique solution of (1.5) in  $\mathcal{C}(\overline{\mathcal{D}}) \cap \mathcal{C}^2(\mathcal{D})$ , which satisfies for all  $x \in \overline{\mathcal{D}}$

$$(1.14) \quad u(x) = \mathbb{E}[F(u, X^x)]$$

where

$$\begin{aligned} F(u, X^x) &= u(X_{\tau^x}^x) \exp \left( - \int_0^{\tau^x} k(X_t^x) dt \right) \\ &\quad + \int_0^{\tau^x} \mathcal{A}(u)(X_t^x) \exp \left( - \int_0^t k(X_s^x) ds \right) dt, \end{aligned}$$

with  $u(X_{\tau^x}^x) = f(X_{\tau^x}^x)$  and  $\mathcal{A}(u)(X_t^x) = g(X_t^x)$ .

*Proof.* We refer the reader to [41] and references therein for a proof of the existence of a classical solution.

Using now the Ito formula for the product of the Ito process  $Y_s$  and the function  $Z_s$  defined by (where  $b^Y$  and  $\sigma^Y$  are scalar drift and diffusion coefficients of the SDE

satisfied by  $Y$ )

$$\begin{aligned} Y_s &= Y_0 + \int_0^s b_r^Y dr + \int_0^s \sigma_r^Y dW_r \\ Z_s &= e^{\int_0^s c_r dr} \end{aligned}$$

we obtain

$$Y_s Z_s = Y_0 + \int_0^s e^{\int_0^r c_{s_1} ds_1} (c_r Y_r + b_r^Y) dr + \int_0^s e^{\int_0^r c_{s_1} ds_1} \sigma_r^Y dW_r$$

Taking  $Y_s = u(X_s^x)$  with  $X_s^x$  the solution process, starting from  $x$  at time 0, of the SDE (1.11) whose infinitesimal generator is  $\mathcal{L}$  (whose drift coefficient is  $b$  and diffusion coefficient is  $\sigma$ ) and  $c_s = -k(X_s^x) \mathbf{1}_{s \geq 0}$  and using the fact that  $u$  is the solution of (1.5)-(1.6), we get

$$\begin{aligned} u(x) &= u(X_s^x) e^{-\int_0^s k(X_r^x) dr} + \int_0^s e^{-\int_0^r k(X_{s_1}^x) ds_1} g(X_r^x) dr \\ &\quad - \int_0^s e^{-\int_0^r k(X_{s_1}^x) ds_1} \nabla_x u(X_r^x) \sigma(X_r^x) dW_r. \end{aligned}$$

Choosing the stopping time as an exit time

$$\tau_n^x = \inf \left\{ s > 0 : d(X_s^x, \partial D) \leq \frac{1}{n} \right\},$$

we have for  $s = \tau_n^x$ , taking at the same time the expectation in the previous formula (the second integral has a zero-valued expectation)

$$u(x) = \mathbb{E} \left( u(X_{\tau_n^x}^x) e^{-\int_0^{\tau_n^x} k(X_r^x) dr} + \int_0^{\tau_n^x} e^{-\int_0^r k(X_{s_1}^x) ds_1} g(X_r^x) dr \right)$$

With  $\tau_n^x \rightarrow \tau^x$  we take the limit and we get (1.14) by continuity on  $\overline{\mathcal{D}}$  of  $u, g$  and  $k$ .  $\square$

Compared to Theorem 1.11, Theorem 1.12 does not assume the existence of a classical solution: this existence is already ensured by the assumptions of the theorem. Among the other assumptions, (E2) ensure the uniqueness of the solution to the SDE, while (E4) ensures that the exit time  $\tau^x$  is almost surely finite for every  $x$  in  $\mathcal{D}$  (see [40, Lemma 3.3.1]).

## 1.3 Numerical Methods

In this section, we first detail the numerical tools we will use to estimate pointwise evaluations of the solution of (1.2)-(1.4) using the Feynman-Kac formula (1.13). It includes time-integration schemes in order to simulate the diffusion process  $X^{t,x}$  and Monte-Carlo methods for expectation estimation. Finally, after a brief analysis of the error resulting from the numerical tools we use. We present in Section 1.3.3 a complexity analysis of these numerical tools together with a survey on techniques reducing this complexity.

### 1.3.1 Approximation of Diffusion Processes

The process  $X^{t,x}$  is traditionally approximated by a time-integration scheme  $S$  designed according to the SDE satisfied by  $X^{t,x}$ . Letting  $t^n = t + n\Delta t$ ,  $n \in \mathbb{N}$ ,  $X^{t,x}$  is approximated by a piecewise constant process  $X^{t,x,\Delta t}$ , where  $X_s^{t,x,\Delta t} = X_{t^n}^{t,x,\Delta t}$  for  $s \in [t^n, t^{n+1}[$  and

$$(1.15) \quad X_{t^{n+1}}^{t,x,\Delta t} = S(X_{t^n}^{t,x,\Delta t}, b, \sigma, t^n, \Delta W_n, \Delta t).$$

Here  $\Delta W_n = W_{t^{n+1}} - W_{t^n}$  is an increment of the standard Brownian motion. In particular the Euler-Maruyama (see Section 2.2.1) time-integration schemes is the most basic way to simulate trajectories of  $X^{t,x,\Delta t}$ .

**Remark 1.13.** *Piecewise constant is not mandatory in the definition of  $X^{t,x,\Delta t}$ . For example in [5] the authors use piecewise affine trajectories (which are continuous). In our analysis we chose piecewise constant approximations for a sake of simplicity.*

Using a time-integration scheme, we can also estimate the exit time  $\tau^{t,x}$  by

$$(1.16) \quad \tau^{t,x} \approx \tau^{t,x,\Delta t} := \min \{t^n > 0 : X_{t^n}^{t,x,\Delta t} \notin \mathcal{D}\}.$$

To quantify the accuracy of the stochastic process  $X^{t,x,\Delta t}$  (one can show that  $X^{t,x,\Delta t}$  is a Ito process, see e.g. [47, Theorem V.1.2]) as an approximation of  $X^{t,x}$  defined by (1.11), we will introduce in Section 1.3.3 the notion of time-integration error.

### 1.3.2 Monte-Carlo Methods

We can use Monte-Carlo approach to approximate the time-integrated solution

$$(1.17) \quad u_{\Delta t}(t, x) = \mathbb{E} \left[ G(u, X^{t,x,\Delta t}) \right].$$

Thus, denoting by  $(X^{t,x,\Delta t}(\omega_m))_{m=1\dots M}$  a family of  $M$  independent trajectories of the stochastic process  $X^{t,x,\Delta t}$  (simulated using time-integration schemes introduced in section 1.3.1) we have

$$(1.18) \quad u_{\Delta t}(t, x) \approx u_{\Delta t,M}(t, x) = \frac{1}{M} \sum_{m=1}^M G(u, X^{t,x,\Delta t}(\omega_m)),$$

where  $G(u, X^{t,x,\Delta t}(\omega_m))$  can be computed by the rectangle quadrature formula with left point (which is exact here since  $X^{t,x,\Delta t}$  is a piecewise constant stochastic process),

$$(1.19) \quad \begin{aligned} G(u, X^{t,x,\Delta t}(\omega_m)) &= f(t^{L(\omega_m)}, X_{t^{L(\omega_m)}}^{t,x,\Delta t}(\omega_m)) \prod_{l=0}^{L(\omega_m)-1} \exp \left( -k(t^l, X_{t^l}^{t,x,\Delta t}) \Delta t \right) \\ &+ \sum_{l=0}^{L(\omega_m)-1} g(t^l, X_{t^l}^{t,x,\Delta t}) \prod_{j=0}^l \exp \left( -k(t^j, X_{t^j}^{t,x,\Delta t}) \Delta t \right), \end{aligned}$$

with  $L$  a random variable such that  $t^L = \tau^{t,x,\Delta t} \wedge T$ .

**Remark 1.14.** *Using the rectangle quadrature formula with right point is possible but when  $t_L(\omega_m) = \tau^{t,x,\Delta t}(\omega_m)$ , some quantities in (1.19) are not well-defined, since  $X_{t_L(\omega_m)}^{t,x,\Delta t}$  is almost surely not in  $\overline{\mathcal{D}}$ , the space domain where  $f, g, k$  are defined.*

### 1.3.3 Statistical Error for the Simulation of Pointwise Evaluations

The estimation error  $\epsilon_{\Delta t, M}$  is defined as

$$(1.20) \quad \epsilon_{\Delta t, M}(t, x) := u(t, x) - u_{\Delta t, M}(t, x).$$

It naturally depends on two parameters, the time-integration discretization parameter  $\Delta t$  and the number of Monte-Carlo samples  $M$ . The error can be decomposed in two terms

$$(1.21) \quad \epsilon_{\Delta t, M}(t, x) = \mathbb{E}[\epsilon_{\Delta t, M}(t, x)] + \epsilon_{\Delta t, M}(t, x) - \mathbb{E}[\epsilon_{\Delta t, M}(t, x)].$$

We notice that  $\mathbb{E}[\epsilon_{\Delta t, M}(t, x)] := u(t, x) - u_{\Delta t}(t, x)$ . This error will be referred to as the time-integration error and be denoted  $\epsilon_{\Delta t}(t, x)$ . The remaining term is due to the Monte-Carlo method and will be noted  $\epsilon_{MC}(t, x)$ . We notice that it corresponds to  $u_{\Delta t}(t, x) - u_{\Delta t, M}(t, x)$  and that its variance is equal to the variance of the estimate.

On one hand, the time-integration error  $\epsilon_{\Delta t}$  is due to the time-integration scheme (1.15) we use. To quantify the precision of the time-integration scheme, we study its strong convergence.

**Definition 1.15** (Strong convergence of time-integrated stochastic process). *A discrete time approximation  $X^{t, x, \Delta t}$  converges strongly with order  $\gamma$ , on  $[t, T]$ , towards  $X^{t, x}$  when there exists a positive constant  $C$ , independent of  $\Delta t$ , such that, for all  $0 < \Delta t \leq \Delta t_{\lim}$ , we have*

$$(1.22) \quad \mathbb{E} \left[ \sup_{t \leq s \leq T} \|X_s^{t, x} - X_s^{t, x, \Delta t}\| \right] \leq C \Delta t^\gamma.$$

Under assumptions slightly stronger than (A1) and (A2) on the coefficients  $b$  and  $\sigma$  of (1.11), it has been proven in [47, Theorem V.2.1] that the Euler-Maruyama scheme (presented in Section 1.2.2 in the stationary case) has a strong order of convergence  $\gamma = \frac{1}{2}$ . Other numerical schemes, assuming stronger conditions on the coefficients of the SDE defining  $X^{t, x}$ , have been proposed to improve the order of convergence. We refer to [71, Chapter 10-12] for explicit time-integration schemes of higher order.

**Remark 1.16.** *An interesting point is that, when the coefficients  $b$  and  $\sigma$  involved in the definition of the infinitesimal generator of  $X^{t, x}$ , see (1.11), are constant, the Euler-Maruyama scheme does not differ from these high-order schemes: the corrective terms added to increase the order of convergence depend on the derivatives of  $b$  and  $\sigma$ . In this situation, other techniques exist to improve the convergence order. We will present them later.*

However, the strong convergence order does not allow to study the behaviour of  $\epsilon_{\Delta t}$  directly. In particular, assumptions on the functional involved in the Feynman-Kac representation have to be made. For example, recalling

$$\epsilon_{\Delta t}(t, x) = \mathbb{E}[G(u, X^{t, x})] - \mathbb{E}[G(u, X^{t, x, \Delta t})]$$

and assuming  $G(u, \cdot)$  is Lipschitz, we can show that  $|\epsilon_{\Delta t}(t, x)| \lesssim \Delta t^{1/2}$  when we use the Euler-Maruyama time-integration scheme. However this is a really naive way to study  $\epsilon_{\Delta t}$ . A more advanced way is often referred to as the weak error study and can be made independently from the strong error.

On the other hand, the Monte-Carlo error  $\epsilon_{MC}$  can be controlled in probability non-asymptotically using concentration inequalities (see [42] in the parabolic case on unbounded domain where  $G(u, X^{t,x,\Delta t}) = f(T, X_T^{t,x,\Delta t})$  for a deterministic value of  $T$ ) which means that for all  $\epsilon > 0$ , it exists  $C$ , not depending on  $\epsilon$  or  $M$ , such that

$$\mathbb{P}(|u_{\Delta t}(t, x) - u_{\Delta t, M}(t, x)| > \epsilon) \leq 2 \exp(-M\epsilon^2 C).$$

From this, we can deduce confidence intervals with length behaving in  $M^{-1/2}$  as expected for Monte-Carlo methods.

In light of these results and the error decomposition (1.21), a balance has to be found between a fine time-discretization parameter  $\Delta t$  and a high number of Monte-Carlo samples  $M$ . This balance depends naturally on the order of convergence of  $\epsilon_{\Delta t}$  with respect to  $\Delta t$  (equal to  $\frac{1}{2}$  here) and also on the order of convergence of  $\epsilon_{MC}$  with  $1/M$  (also equal to  $\frac{1}{2}$  here). If we denote by  $\epsilon$  the target accuracy we aim at reaching for our pointwise evaluations, we have to take  $M = O(\epsilon^{-2})$  and  $\Delta t = O(\epsilon^2)$ , due to the order of convergence of the time-integration error and the Monte-Carlo error that are in both cases equal to  $\frac{1}{2}$ . The complexity being  $O(M\Delta t^{-1})$ , as function of the wanted precision it behaves like  $O(\epsilon^{-4})$ . Thus, for very precise pointwise evaluations, the complexity becomes too high. A solution to overcome these accuracy limitations is to improve the order of convergence of both the time-integration technique and the Monte-Carlo method. We recall hereafter a few ways to enhance these orders. In Section 1.3.3.1 we survey high-order time-integration methods while Section 1.3.3.2 is devoted to enhanced Monte-Carlo methods.

### 1.3.3.1 Time-Integration Error $\epsilon_{\Delta t}$

It has been proven in [47, Theorem V.3.1] that for a parabolic Cauchy problem on unbounded domains, i.e. when no exit-time has to be estimated, the time-integration error with an Euler-Maruyama scheme behaved as  $O(\Delta t)$ . Combined with classical Monte-Carlo methods this would allow an overall complexity in  $O(\epsilon^{-3})$ . However, the assumptions on the coefficients of the PDE were very strict to obtain this result ( $\mathcal{C}^{1,4}([0, T] \times \overline{\mathcal{D}})$ , plus the existence of a classical solution with the same regularity). For a simpler but classical study case (parabolic on unbounded domain with  $g \equiv k \equiv 0$ ), the time-integration error  $\mathbb{E}[f(X_T^{t,x})] - \mathbb{E}[f(X_T^{t,x,\Delta t})]$  has been studied in [99, 5] while using the Euler-Maruyama time-integration scheme. In [99], taking  $f$  and the coefficients of the SDE driving  $X^{t,x}$  to be smooth functions, the authors give an expansion of the error in term of powers of  $\Delta t$

$$(1.23) \quad \mathbb{E}[f(X_T^{t,x})] - \mathbb{E}[f(X_T^{t,x,\Delta t})] = \sum_{r=1}^R a_r \Delta t^r + O(\Delta t^{R+1}),$$

where the coefficients  $(a_r)_{r=1}^{R+1}$  do not depend on  $\Delta t$ . In [5], a similar expansion is obtained when  $f$  is only measurable and under non-degeneracy assumption on the infinitesimal generator of  $X^{t,x}$ . Such expansions are then useful to propose new approximations (of higher order) for  $\mathbb{E}[f(X_T^{t,x})]$  by the principle of Romberg extrapolation, that we present here when  $R = 1$  in (1.23). In this case we can notice that

$$(1.24) \quad \mathbb{E}[f(X_T^{t,x})] - \left(2\mathbb{E}[f(X_T^{t,x,\Delta t/2})] - \mathbb{E}[f(X_T^{t,x,\Delta t})]\right) = O(\Delta t^2),$$

and then we have a time-integrated approximation whose order of convergence is 2. This technique can be extended to any order  $R + 1$  if we have an expansion of order  $R$ . For more generalities on the subject, see for example [99, Section 2.3] or [89].

**Remark 1.17.** *Such a technique can be used to propose high-order approximation in the case where the coefficients  $b$  and  $\sigma$  are constant (see Remark 1.16).*

These techniques, even if interesting in practice, cannot be used for any function  $f$  (for which the existence of an expansion might be harder to prove), and we still have to use time-integration schemes with time-integration parameter lower than  $\Delta t$  ( $\Delta t/2$  for example when using the Romberg extrapolation technique to get an order 2 approximation), which is numerically costly.

Another major concern, not treated by the already presented literature, is the exit-time estimation which introduces also a bias: we easily notice that  $\tau^{t,x,\Delta t} \geq \tau^{t,x}$  which introduces a bias in  $O(\sqrt{\Delta t})$  according to [50, Theorem 4]. To deal with this overestimation it is proposed in [50, Theorem 5] to consider a slightly modified exit-time to regain a time-integration error in  $o(\sqrt{\Delta t})$ : this exit time is

$$\tau_1^{t,x,\Delta t} := \inf\{s > t : X^{t,x,\Delta t} \notin \mathcal{D}^{\Delta t}\},$$

where  $\mathcal{D}^{\Delta t} = \{x \in \mathcal{D} : d(x, \partial\mathcal{D}) > C\sqrt{\Delta t}\}$  is a smaller domain of  $\mathcal{D}$ , see Figure 1.1. This technique is very useful in practice since it is easy to implement. However the resulting convergence has only been proved to behave like  $o(\Delta t^{1/2})$ , the exact order has not been explicit yet.

In the general parabolic case (1.2)-(1.4), high order of convergence are difficult to reach using traditional techniques without very restrictive assumptions on the PDE. This is due to the complicated form of the Feynman-Kac functional in (1.13). However, for easier study cases, interesting alternatives can be proposed. In fact one can notice that the simulation of stochastic processes' trajectories is complicated. If we were considering problems for which the simulation of a whole trajectory is not needed, we could investigate better precision without higher numerical complexity. Here we consider two different elliptic problems (1.5)-(1.6) for which we do not need the simulation of the entire trajectory. A first one where  $k \equiv b \equiv g \equiv 0$ ,  $f \equiv \sigma \equiv 1$ . In this simplified case, known as the Poisson problem, the Feynman-Kac formula becomes

$$(1.25) \quad u(x) = \mathbb{E}[f(X_{\tau^x}^x)],$$

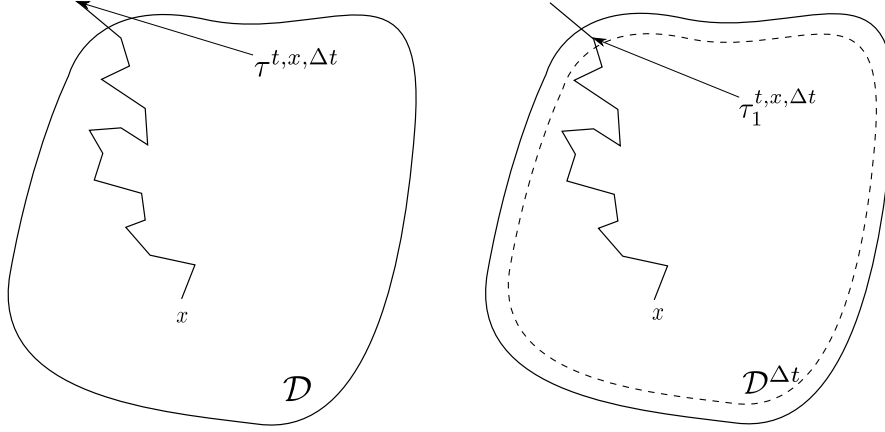


Figure 1.1: Comparison between the estimation of the exit time  $\tau^{t,x,\Delta t}$  of  $\mathcal{D}$  and the estimation of the exit time  $\tau_1^{t,x,\Delta t}$  of  $\mathcal{D}^{\Delta t}$  [50], illustrated for a particular trajectory of  $X^{t,x,\Delta t}$ . The domain  $\mathcal{D}^{\Delta t}$  (delimited by - -) allows to estimate an exit position (and exit time) with a bias lower than the one using  $\mathcal{D}$  (delimited by —).

where  $\tau^x = \tau^{0,x}$  by convention. This problem consist here in estimating the exit point of the domain  $\mathcal{D}$ . A second one where  $k \equiv b \equiv 0$ ,  $f \equiv g \equiv \sigma \equiv 1$ . The Feynman-Kac formula thus becomes

$$(1.26) \quad u(x) = \mathbb{E}[f(X_{\tau^x}^x) + \tau^x].$$

Here we have to estimate both the exit point and the exit time of the domain  $\mathcal{D}$ .

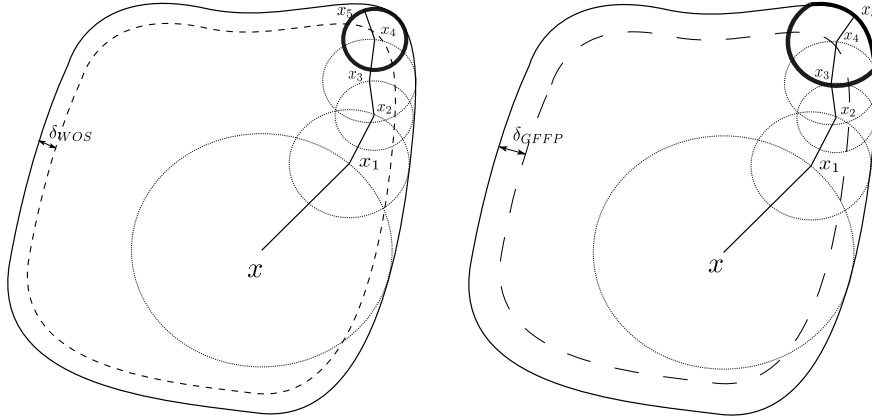


Figure 1.2: Comparison between Walk on Sphere (left) and Green Function First Passage (right) techniques [62]. Both procedures work the same, until the selection of  $x_5$ . When  $x_4$  has been drawn, it lays at a distance to the boundary smaller than  $\delta_{GFFP}$ . So we can use the GFFP method and draw the next point on the intersection of a sphere and the boundary (in bold on the left). The WOS draws the next sample still on a sphere (in bold on the right) and stops since the next sample  $x_5$  lays at a distance lower than  $\delta_{WOS}$  of the boundary.

A groundbreaking technique has been theoretically introduced in [82] to solve (1.25): the Walk-On-Sphere (WOS) procedure. It is based on the observation that the exit position of a sphere, for a Brownian motion starting in the center of this sphere, is uniformly distributed over the sphere. Then, consider procedure described here



1. Start in  $x_0 = x$  and set  $k = 0$ .
2. Find the largest sphere, centered in  $x_k$ , contained in  $\mathcal{D}$ . Draw according to the uniform distribution a point  $x_{k+1}$  on this sphere. If this point is on  $\partial\mathcal{D}$ , stop.
3. Otherwise, increment  $k$  and repeat the procedure from step 2.

It returns a point on  $\partial\mathcal{D}$  and this point has the same distribution as  $X_{\tau^x}^x$ . However, this procedure almost surely does not stop. Simply modifying it by stopping when the point  $x_{k+1}$  is closer than some parameter  $\delta_{WOS}$  from the boundary, [80] analyzes the numerical error resulting from this technique. It gives then a result on the average number  $N(\delta_{WOS})$  of spheres before stopping: it is bounded, up to a constant, by  $|\log(\delta_{WOS})|$ . Under regularity assumptions on the boundary, the error of such method is in  $O(\delta_{WOS})$  [76]. Thus, the complexity (to get one sample) as a function of the precision  $\epsilon$  behaves in  $O(\log(\epsilon^{-1}))$  which is better than techniques whose complexity is a power of  $\epsilon^{-1}$ . The WOS procedure has then been adapted to rectangular domains in [29]. It has been later improved under the name of Green Function First Passage (GFFP) [45] in order to avoid the absorption layer of size  $\delta_{WOS}$  and then to simulate exactly  $\tau^x$ . An illustrated comparison of GFFP and WOS methods can be found on Figure 1.2, see [62] for a comparison in term of complexity of the two methods. This GFFP method thus requires the exact simulation of the exit time over domains intersected with spheres (see Figure 1.2). An improvement of GFFP, called the Simulation-Tabulation method has even been proposed in [63]. These techniques are however difficult to extend to non-classical domain  $\mathcal{D}$  (or to high dimensional problems on complicated domain) since it is necessary to get samples according to the distribution of the exit time of a Brownian motion from domain more complicated than spheres (see [45, Appendix] among other references). Moreover, it is only available for the Poisson equation, when the diffusion process is in fact the Brownian motion.

All these techniques to estimate the exit position from a bounded domain of the Brownian motion have been completed more recently by the study of the exit time of other one-dimensional stochastic processes [57, 58, 59]. This new class of algorithms, made to estimate (1.26), are called the Walk-On-Moving-Spheres (WOMS) algorithms [30].

Another field of interest consists in studying PDEs with non-regular coefficients, which differs from the classical presentation adopted in Theorem 1.11 or Theorem 1.12. We refer the reader to [15] and references therein for an overview for estimating (1.26) with such coefficients.

Promising numerical results allow to exactly simulate trajectories of diffusion processes in the case  $d = 1$  [9, 34, 35]. They are presented as allowing to reach higher precision for the same numerical complexity. However, up to our knowledge, such methods have not yet been employed for solving PDEs.

As a conclusion for this part, we summarize all the literature presented above in Table 1.1.

Problem studied	Technique used	Advantages/Drawbacks	Complexity
(1.14) or (1.13)	Euler-Maruyama scheme	Easy to implement and to apply to any diffusion process <b>but</b> low convergence order	$O(\epsilon^{-1})$ or $O(\epsilon^{-2})$ depending on the regularity for the functions in (1.14)
	High-order schemes [71]	High convergence order compared to Euler-Maruyama scheme <b>but</b> not available when the SDE has constant coefficients, less easy to implement in high-dimension	$O(\epsilon^{-\gamma})$ , $\gamma \in (0, 1]$ depending on the order of the scheme
	Boundary shifting [50]	Easy to implement	$O(\epsilon^{-1})$
Poisson equation on unbounded domain	Romberg extrapolation [99, section 2.3]	Allow high-order of convergence even when the SDE has constant coefficients <b>but</b> difficult to extend to complex function of $X^{t,x}$	$O(\epsilon^{-\gamma})$ , $\gamma \in (0, 1]$ depending on the order of the expansion
(1.25), small $d$	WOS and GFFP [62]	High convergence order <b>but</b> difficult to extend to complex or high-dimensional space domain or to other diffusion processes	$O(\log(\epsilon^{-1}))$
(1.26), small $d$	WOMS [30] or [58, 59]	High convergence order <b>but</b> difficult to extend to more complex or high-dimensional space domain or to other diffusion processes	$O(\log(\epsilon^{-1}))$
$(X_s^{t,x})_{0 \leq s \leq T}$ , $d = 1$	[34, 35]	Exact simulation of trajectories <b>but</b> the diffusion coefficient has to be equal to 1	Lower complexity at given precision compared to standard simulation of trajectories.

Table 1.1: Summary of the advantages and drawbacks of each numerical time-integration technique. The complexity column corresponds to the complexity to reach a precision of  $\epsilon$  for  $\epsilon_{\Delta t}$ .

### 1.3.3.2 Monte-Carlo Error $\epsilon_{MC}$

Regarding the Monte-Carlo error  $\epsilon_{MC}$ , the order of convergence of the error, with respect to the number of Monte-Carlo samples  $M$ , behaves in  $O(M^{-1/2})$ . This order is difficult to reduce, except using multilevel Monte-Carlo methods that we will present later. However it is possible to improve the estimate  $u_{\Delta t, M}(t, x)$  by using variance reduction techniques. In order to see the interest of these techniques, we introduce

$$(1.27) \quad \mathbb{V}[u_{\Delta t, M}(t, x)] = \frac{\mathbb{V}[G(u, X^{t,x, \Delta t})]}{M}.$$

When the variance  $\mathbb{V}[G(u, X^{t,x, \Delta t})]$  is large, the estimation using the classic Monte-Carlo estimate is not adapted for precise estimation of  $u(t, x)$ . It is thus possible

to use variance reduction techniques, e.g. antithetic sampling. Suppose that  $X^{t,x,\Delta t}$  and  $\tilde{X}^{t,x,\Delta t}$  have the same law. In particular since their expectations are equal, we have

$$\begin{aligned} u_{\Delta t}(t, x) &= \mathbb{E}[G(u, X^{t,x,\Delta t})] \\ &= \mathbb{E}[G(u, \tilde{X}^{t,x,\Delta t})] = \frac{\mathbb{E}[G(u, X^{t,x,\Delta t}) + G(u, \tilde{X}^{t,x,\Delta t})]}{2} \end{aligned}$$

Using the last equality, it is possible to propose the following estimate

$$(1.28) \quad \tilde{u}_{\Delta t, M}(t, x) = \frac{2}{M} \sum_{m=1}^{M/2} \frac{G(u, X^{t,x,\Delta t}(\omega_m)) + G(u, \tilde{X}^{t,x,\Delta t}(\tilde{\omega}_m))}{2},$$

where the  $X^{t,x,\Delta t}(\omega_m)$  are i.i.d. samples of  $X^{t,x,\Delta t}$  and the  $\tilde{X}^{t,x,\Delta t}(\tilde{\omega}_m)$  are i.i.d. samples of  $\tilde{X}^{t,x,\Delta t}$ . This estimate is better than  $u_{\Delta t, M}(t, x)$  when we can prove that the covariance  $\text{Cov}[G(u, X^{t,x,\Delta t}), G(u, \tilde{X}^{t,x,\Delta t})] < 0$ . This variance reduction technique is particularly relevant and easy to implement in some cases here since when  $X^{t,x,\Delta}$  is the standard discretized Brownian motion, then so is  $-X^{t,x,\Delta}$ . Other variance reduction techniques exist and we refer the reader to [47, Chapter III] for a complete presentation.

We focus now on multilevel methods [44]. In some cases, these techniques also allow to improve the convergence rate of  $\epsilon_{MC}$  with reference to  $M$ . If we want to estimate  $\mathbb{E}[Z_L]$  but  $Z_L$  is costly to sample, it might be better, if we have access to an approximation  $Z_{L-1}$  of  $Z_L$  which is less costly to sample but also less precise, to use the following equality

$$\mathbb{E}[Z_L] = \mathbb{E}[Z_{L-1}] + \mathbb{E}[Z_L - Z_{L-1}],$$

in order to propose the following estimate

$$(1.29) \quad \hat{\mathbb{E}}[Z_L] = \frac{\sum_{m=1}^{M_1} Z_{L-1}(\omega_m^1)}{M_1} + \frac{\sum_{m=1}^{M_2} Z_L(\omega_m^2) - Z_{L-1}(\omega_m^2)}{M_2},$$

where the  $Z_j(\omega^i)$  are independent samples drawn according to the law of  $Z_j$ . It is important to make sure that the samples used in the difference  $Z_L(\omega_m^2) - Z_{L-1}(\omega_m^2)$  are drawn according to the same underlying elementary event  $\omega_m^2$ . This multilevel Monte-Carlo method with two levels can then be generalized to a  $L$  level multilevel Monte-Carlo method using the following equality

$$(1.30) \quad \mathbb{E}[Z_L] = \sum_{l=0}^{L-1} \mathbb{E}[Z_{L-l} - Z_{L-l-1}], \quad Z_0 = 0,$$

just assuming that we have access to a sequence of approximation with decreasing cost and precision  $(Z_{L-l})_{l=1}^L$ . Multilevel methods are particularly relevant for the simulation of Feynman-Kac representation, e.g. [61], since it is easy to have access to the different levels required to use them: each level  $l$  corresponds to a Euler-Maruyama time-integration scheme of parameter  $\Delta t_l$ , with  $\Delta t_l$  that decreases with  $l$ . Of course an optimization has to be made for the choice of the number of samples

$M_l$  to be used at each level  $l$ , as well as the choice of the different  $Z_l$ , i.e. of  $\Delta t_l$ , when this choice can be driven. Improved non-asymptotic confidence intervals have been proposed for the multilevel method [64]. The Monte-Carlo error  $\epsilon_{MC}$  using these multilevel methods behaves at most and under additional assumptions, as a function of the reached precision  $\epsilon$ , in  $O(\epsilon \log(\epsilon^{-1}))$  (see among others [47, Theorem IV.3.1]).



# Chapter 2

## Probabilistic Methods for Solving High-Dimensional Elliptic PDEs

This chapter introduced our first contribution. It consists in an algorithm to approximate up to any precision the solution of a high-dimensional elliptic partial differential equation (PDE) on a bounded domain.

### Contents

2.1	Introduction . . . . .	46
2.2	Probabilistic Tools for Solving PDEs . . . . .	47
2.2.1	Pointwise Evaluations of the Solution . . . . .	47
2.2.2	A Sequential Control Variates Algorithm . . . . .	50
2.3	Adaptive Sparse Interpolation . . . . .	52
2.3.1	Sparse Interpolation . . . . .	52
2.3.2	Adaptive Algorithm for Sparse Interpolation . . . . .	53
2.4	Main Contributions . . . . .	54
2.4.1	Perturbed Version of Sparse Interpolation . . . . .	54
2.4.2	Adaptive Version of Sequential Control Variate Algorithm . . . . .	54
2.4.3	Numerical Results . . . . .	55
2.5	Conclusion . . . . .	59

As we have seen in Section 1.3.3, the complexity as a function of the reached precision  $\epsilon$  behaves polynomially in  $\epsilon^{-1}$ . Taking a closer look at this complexity, reaching low precision, e.g. machine precision, through these methods would cost a high number of Monte-Carlo samples together with time-integration schemes with very fine parameter  $\Delta t$ . While it may be possible to conduct such techniques for a single pointwise evaluation, such an approach are not relevant in a multi-query context, which is when several pointwise evaluations are required. Particularly since deterministic methods are relevant in this context for low-dimensional problems, i.e. when  $d$  is small.

Moreover the most efficient time-integration techniques presented in Section 1.3.3 mainly for low-dimensional problems are difficult to implement or to extend to high-dimensional problems. Thus it is particularly costly to reach machine precision for

pointwise evaluations of high-dimensional problems.

In this chapter, we present work that has already been published in [11]. We consider high-dimensional PDEs, i.e.  $d \gg 1$ , and in the multi-query context. We propose two algorithms combining global approximation techniques with the probabilistic methods for PDEs seen in Chapter 1. These algorithms provide heuristics allowing to reach machine precision while approximating the solution of a high-dimensional PDE on bounded domain.

---

**Title:** Probabilistic methods for solving high-dimensional PDEs.

**Authors:** Marie Billaud-Friess ♣, Arthur Macherey ♣♠, Anthony Nouy ♣ and Clémentine Prieur ♠.

♣ Centrale Nantes, LMJL, UMR CNRS 6629, 1 rue de la Noë, 44321 Nantes

♠ Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP\*, LJK, 38000 Grenoble, France

\*Institute of Engineering Univ. Grenoble Alpes

**Abstract:** We propose algorithms for solving high-dimensional partial differential equations (PDEs) that combine a probabilistic interpretation of PDEs, through Feynman-Kac representation, with sparse interpolation. Monte-Carlo methods and time-integration schemes are used to estimate pointwise evaluations of the solution of a PDE. We use a sequential control variates algorithm, where control variates are constructed based on successive approximations of the solution of the PDE. Two different algorithms are proposed, combining in different ways the sequential control variates algorithm and adaptive sparse interpolation. Numerical examples will illustrate the behavior of these algorithms.

---

## 2.1 Introduction

We consider the solution of an elliptic partial differential equation

$$(2.1) \quad \begin{aligned} \mathcal{A}(u) &= g \quad \text{in } \mathcal{D}, \\ u &= f \quad \text{on } \partial\mathcal{D}, \end{aligned}$$

where  $u : \overline{\mathcal{D}} \rightarrow \mathbb{R}$  is a real-valued function, and  $\mathcal{D}$  is an open bounded domain in  $\mathbb{R}^d$ .  $\mathcal{A}$  is an elliptic linear differential operator and  $f : \partial\mathcal{D} \rightarrow \mathbb{R}$ ,  $g : \overline{\mathcal{D}} \rightarrow \mathbb{R}$  are respectively the boundary condition and the source term of the PDE.

We are interested in approximating the solution of (2.1) up to a given precision. For high dimensional PDEs ( $d \gg 1$ ), this requires suitable approximation formats such as sparse tensors [19, 94] or low-rank tensors [87, 52, 55, 4, 84]. Also, this requires algorithms that provide approximations in a given approximation format. Approximations are typically provided by Galerkin projections using variational formulations of PDEs. Another path consists in using a probabilistic representation of the solution  $u$  through Feynman-Kac formula, and Monte-Carlo methods to provide estimations of pointwise evaluations of  $u$  (see e.g., [47]). This allows to compute

approximations in a given approximation format through classical interpolation or regression [8, 101, 7]. In [48, 49], the authors consider interpolations on fixed polynomial spaces and propose a sequential control variates method for improving the performance of Monte-Carlo estimation. In this paper, we propose algorithms that combine this variance reduction method with adaptive sparse interpolation [21, 22].

The outline is as follows. In Section 2.2, we recall the theoretical and numerical aspects associated to probabilistic tools for estimating the solution of (2.1). We also present the sequential control variates algorithm introduced in [48, 49]. In Section 2.3 we introduce sparse polynomial interpolation methods and present a classical adaptive algorithm. In Section 2.4, we present two algorithms combining the sequential control variates algorithm from Section 2.2 and adaptive sparse polynomial interpolation. Finally, numerical results are presented in Section 2.4.

## 2.2 Probabilistic Tools for Solving PDEs

We consider the problem (2.1) with a linear partial differential operator defined by  $\mathcal{A}(u) = -\mathcal{L}(u) + ku$ , where  $k$  is a real valued function defined on  $\overline{\mathcal{D}}$ , and where

$$(2.2) \quad \mathcal{L}(u)(x) = \frac{1}{2} \sum_{i,j=1}^d (\sigma(x)\sigma(x)^T)_{ij} \partial_{x_i x_j}^2 u(x) + \sum_{i=1}^d b_i(x) \partial_{x_i} u(x)$$

is the infinitesimal generator associated to the  $d$ -dimensional diffusion process  $X^x$  solution of the stochastic differential equation

$$(2.3) \quad dX_t^x = b(X_t^x)dt + \sigma(X_t^x)dW_t, \quad X_0^x = x \in \overline{\mathcal{D}},$$

where  $W$  is a  $d$ -dimensional Brownian motion and  $b := (b_1, \dots, b_d)^T : \mathbb{R}^d \rightarrow \mathbb{R}^d$  and  $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times d}$  stand for the drift and the diffusion respectively.

### 2.2.1 Pointwise Evaluations of the Solution

The following theorem recalls the Feynman-Kac formula (see [28, Theorem 2.4] or [41, Theorem 2.4] and the references therein) that provides a probabilistic representation of  $u(x)$ , the solution of (2.1) evaluated at  $x \in \overline{\mathcal{D}}$ .

**Theorem 2.1** (Feynman-Kac formula for elliptic PDE on bounded domain). *Assume that*

(E1)  $\mathcal{D}$  is an open connected bounded domain of  $\mathbb{R}^d$ , regular in the sense that, if  $\tau^x = \inf \{s > 0 : X_s^x \notin \mathcal{D}\}$  is the first exit time of  $\mathcal{D}$  for the process  $X^x$ , solution of (2.3), we have

$$\mathbb{P}(\tau^x = 0) = 1, \quad x \in \partial\mathcal{D},$$

(E2)  $b, \sigma$  are Lipschitz functions over  $\overline{\mathcal{D}}$ ,

(E3)  $f$  is continuous on  $\partial\mathcal{D}$ ,  $g$  and  $k \geq 0$  are Hölder-continuous functions on  $\overline{\mathcal{D}}$ ,



(E<sub>4</sub>) (uniform ellipticity assumption) there exists  $c > 0$  such that

$$\sum_{i,j=1}^d \left( \sigma(x) \sigma(x)^T \right)_{ij} \Upsilon_i \Upsilon_j \geq c \sum_{i=1}^d \Upsilon_i^2, \quad \Upsilon \in \mathbb{R}^d, \quad x \in \overline{\mathcal{D}}.$$

Then, there exists a unique solution of (2.1) in  $\mathcal{C}(\overline{\mathcal{D}}) \cap \mathcal{C}^2(\mathcal{D})$ , which satisfies for all  $x \in \overline{\mathcal{D}}$

$$(2.4) \quad u(x) = \mathbb{E} [F(u, X^x)]$$

where

$$\begin{aligned} F(u, X) &= u(X_{\tau^x}^x) \exp \left( - \int_0^{\tau^x} k(X_t^x) dt \right) \\ &\quad + \int_0^{\tau^x} \mathcal{A}(u)(X_t^x) \exp \left( - \int_0^t k(X_s^x) ds \right) dt, \end{aligned}$$

with  $u(X_{\tau^x}^x) = f(X_{\tau^x}^x)$  and  $\mathcal{A}(u)(X_t^x) = g(X_t^x)$ .

Note that  $F(u, X^x)$  in (2.4) only depends on the values of  $u$  on  $\partial D$  and  $\mathcal{A}(u)$  on  $D$ , which are the given data  $f$  and  $g$  respectively. A Monte-Carlo method can then be used to estimate  $u(x)$  using (2.4), which relies on the simulation of independent samples of an approximation of the stochastic process  $X^x$ . This process is here approximated by an Euler-Maruyama scheme. More precisely, letting  $t^n = n\Delta t$ ,  $n \in \mathbb{N}$ ,  $X^x$  is approximated by a piecewise constant process  $X^{x,\Delta t}$ , where  $X_t^{x,\Delta t} = X_{t^n}^{x,\Delta t}$  for  $t \in [t^n, t^{n+1}[$  and

$$(2.5) \quad \begin{aligned} X_{t^{n+1}}^{x,\Delta t} &= X_{t^n}^{x,\Delta t} + \Delta t \, b(X_{t^n}^{x,\Delta t}) + \sigma(X_{t^n}^{x,\Delta t}) \, \Delta W_n, \\ X_0^{x,\Delta t} &= x. \end{aligned}$$

Here  $\Delta W_n = W_{t^{n+1}} - W_{t^n}$  is an increment of the standard Brownian motion. For details on time-integration schemes, the reader can refer to [71]. Letting  $\{X^{x,\Delta t}(\omega_m)\}_{m=1}^M$  be independent samples of  $X^{x,\Delta t}$ , we obtain an estimation  $u_{\Delta t, M}(x)$  of  $u(x)$  defined as

$$(2.6) \quad \begin{aligned} u_{\Delta t, M}(x) &:= \frac{1}{M} \sum_{m=1}^M F(u, X^{x,\Delta t}(\omega_m)) \\ &= \frac{1}{M} \sum_{m=1}^M \left[ f(X_{\tau^{x,\Delta t}}^{x,\Delta t}(\omega_m)) \exp \left( - \int_0^{\tau^{x,\Delta t}} k(X_t^{x,\Delta t}(\omega_m)) dt \right) \right. \\ &\quad \left. + \int_0^{\tau^{x,\Delta t}} g(X_t^{x,\Delta t}(\omega_m)) \exp \left( - \int_0^t k(X_s^{x,\Delta t}(\omega_m)) ds \right) dt \right] \end{aligned}$$

where  $\tau^{x,\Delta t}$  is the first exit time of  $D$  for the process  $X^{x,\Delta t}(\omega_m)$ , given by

$$\tau^{x,\Delta t} = \inf \{ t > 0 : X_t^{x,\Delta t} \notin \mathcal{D} \} = \min \{ t_n > 0 : X_{t_n}^{x,\Delta t} \notin \mathcal{D} \}.$$

**Remark 2.2.** *In practice,  $f$  has to be defined over  $\mathbb{R}^d$  and not only on the boundary  $\partial\mathcal{D}$ . Indeed, although  $X_{\tau^x}^x \in \partial\mathcal{D}$  with probability one,  $X_{\tau^x, \Delta t}^{x, \Delta t} \in \mathbb{R}^d \setminus \overline{\mathcal{D}}$  with probability one.*

The error can be decomposed in two terms

$$(2.7) \quad u(x) - u_{\Delta t, M}(x) = \overbrace{u(x) - \mathbb{E} \left[ F(u, X^{x, \Delta t}) \right]}^{\varepsilon_{\Delta t}} + \underbrace{\mathbb{E} \left[ F(u, X^{x, \Delta t}) \right] - \frac{1}{M} \sum_{m=1}^M F(u, X^{x, \Delta t}(\omega_m))}_{\varepsilon_{MC}},$$

where  $\varepsilon_{\Delta t}$  is the time integration error and  $\varepsilon_{MC}$  is the Monte-Carlo estimation error. Before discussing the contribution of each of both terms to the error, let us introduce the following additional assumption, which ensures that  $\mathcal{D}$  does not have singular points<sup>[1]</sup>.

- (E5) Each point of  $\partial\mathcal{D}$  satisfies the exterior cone condition which means that, for all  $x \in \partial\mathcal{D}$ , there exists a finite right circular cone  $K$ , with vertex  $x$ , such that  $\overline{K} \cap \overline{\mathcal{D}} = \{x\}$ .

Under assumptions (E1)-(E5), it can be proven [49, §4.1] that the time integration error  $\varepsilon_{\Delta t}$  converges to zero. It can be improved to  $O(\Delta t^{1/2})$  by adding differentiability assumptions on the boundary [50]. The estimation error  $\varepsilon_{MC}$  is a random variable with zero mean and standard deviation converging as  $O(M^{-1/2})$ . The computational complexity for computing a pointwise evaluation of  $u_{\Delta t, M}(x)$  is in  $O(M\Delta t^{-1})$  in expectation for  $\Delta t$  sufficiently small<sup>[2]</sup>, so that the computational complexity for achieving a precision  $\epsilon$  (root mean squared error) behaves as  $O(\epsilon^{-4})$ . This does not allow to obtain a very high accuracy in a reasonable computational time. The convergence with  $\Delta t$  can be improved to  $O(\Delta t)$  by suitable boundary corrections [50], therefore yielding a convergence in  $O(\epsilon^{-3})$ . To further improve the convergence, high-order integration schemes could be considered (see [71] for a survey). Also, variance reduction methods can be used to further improve the convergence, such as antithetic variables, importance sampling, control variates (see [47]). Multilevel Monte-Carlo [44] can be considered as a variance reduction method using several control variates (associated with processes  $X^{x, \Delta t_k}$  using different time discretizations). Here, we rely on the sequential control variates algorithm proposed in [48] and analyzed in [49]. This algorithm constructs a sequence of approximations of  $u$ . At each iteration of the algorithm, the current approximation is used as a control variate for the estimation of  $u$  through Feynman-Kac formula.

<sup>[1]</sup>Note that together with (E2), assumption (E5) implies (E1) (see [49, §4.1] for details), so that the set of hypotheses (E1)-(E5) could be reduced to (E2)-(E5).

<sup>[2]</sup>A realization of  $X^{x, \Delta t}$  over the time interval  $[0, \tau^{x, \Delta t}]$  can be computed in  $O(\tau^{x, \Delta t} \Delta t^{-1})$ . Then, the complexity to evaluate  $u_{\Delta t, M}(x)$  is in  $O(\mathbb{E}(\tau^{x, \Delta t}) M \Delta t^{-1})$  in expectation. Under (E1)-(E5), it is stated in the proof of [49, Theorem 4.2] that  $\sup_x \mathbb{E}[\tau^{x, \Delta t}] \leq C$  with  $C$  independent of  $\Delta t$  for  $\Delta t$  sufficiently small.

### 2.2.2 A Sequential Control Variates Algorithm

Here we recall the sequential control variates algorithm introduced in [48] in a general interpolation framework. We let  $V_\Lambda \subset \mathcal{C}^2(\overline{\mathcal{D}})$  be an approximation space of finite dimension  $\#\Lambda$  and let  $\mathcal{I}_\Lambda : \mathbb{R}^{\mathcal{D}} \rightarrow V_\Lambda$  be the interpolation operator associated with a unisolvent grid  $\Gamma_\Lambda = \{x_\nu : \nu \in \Lambda\}$ . We let  $(l_\nu)_{\nu \in \Lambda}$  denote the (unique) basis of  $V_\Lambda$  that satisfies the interpolation property  $l_\nu(x_\mu) = \delta_{\nu\mu}$  for all  $\nu, \mu \in \Lambda$ . The interpolation  $\mathcal{I}_\Lambda(w) = \sum_{\nu \in \Lambda} w(x_\nu) l_\nu(x)$  of function  $w$  is then the unique function in  $V_\Lambda$  such that

$$\mathcal{I}_\Lambda(w)(x_\nu) = w(x_\nu), \quad \nu \in \Lambda.$$

The following algorithm provides a sequence of approximations  $(\tilde{u}^k)_{k \geq 1}$  of  $u$  in  $V_\Lambda$ , which are defined by  $\tilde{u}^k = \tilde{u}^{k-1} + \tilde{e}^k$ , where  $\tilde{e}^k$  is an approximation of  $e^k$ , solution of

$$\begin{aligned} \mathcal{A}(e^k)(x) &= g(x) - \mathcal{A}(\tilde{u}^{k-1})(x), & x \in \mathcal{D}, \\ e^k(x) &= f(x) - \tilde{u}^{k-1}(x), & x \in \partial\mathcal{D}. \end{aligned}$$

Note that  $e^k$  admits a Feynman-Kac representation  $e^k(x) = \mathbb{E}(F(e^k, X^x))$ , where  $F(e^k, X^x)$  depends on the residuals  $g - \mathcal{A}(\tilde{u}^{k-1})$  on  $\mathcal{D}$  and  $f - \tilde{u}^{k-1}$  on  $\partial\mathcal{D}$ . The approximation  $\tilde{e}^k$  is then defined as the interpolation  $\mathcal{I}_\Lambda(e_{\Delta t, M}^k)$  of the Monte-Carlo estimate  $e_{\Delta t, M}^k(x)$  of  $e_{\Delta t}^k(x) = \mathbb{E}(F(e^k, X^{x, \Delta t}))$  (using  $M$  samples of  $X^{x, \Delta t}$ ). For

---

**Algorithm 2.1** Sequential Control Variates Algorithm [49]

---

- 1: Set  $\tilde{u}^0 = 0$ ,  $k = 1$  and  $S = 0$ .
  - 2: **while**  $k \leq K$  and  $S < n_s$  **do**
  - 3:   Compute  $e_{\Delta t, M}^k(x_\nu)$  for  $x_\nu \in \Gamma_\Lambda$ .
  - 4:   Compute  $\tilde{e}^k = \mathcal{I}_\Lambda(e_{\Delta t, M}^k) = \sum_{\nu \in \Lambda} e_{\Delta t, M}^k(x_\nu) l_\nu(x)$ .
  - 5:   Update  $\tilde{u}^k = \tilde{u}^{k-1} + \tilde{e}^k$ .
  - 6:   If  $\|\tilde{u}^k - \tilde{u}^{k-1}\|_2 \leq \epsilon_{tol} \|\tilde{u}^{k-1}\|_2$  then  $S = S + 1$  else  $S = 0$ .
  - 7:   Set  $k = k + 1$ .
  - 8: **end while**
- 

practical reasons, Algorithm 2.1 is stopped using an heuristic error criterion based on stagnation. This criterion is satisfied when the desired tolerance  $\epsilon_{tol}$  is reached for  $n_s$  successive iterations (in practice we chose  $n_s = 5$ ).

Now let us provide some convergence results for Algorithm 2.1. To that goal, we introduce the time integration error at point  $x$  for a function  $h$

$$(2.8) \quad e^{\Delta t}(h, x) = \mathbb{E}[F(h, X^{x, \Delta t})] - \mathbb{E}[F(h, X^x)].$$

Then the following theorem [49, Theorem 3.1] gives a control of the error in expectation.

**Theorem 2.3.** *Assuming (E2)-(E5), it holds*

$$\sup_{\nu \in \Lambda} \left| \mathbb{E} [\tilde{u}^{n+1}(x_\nu) - u(x_\nu)] \right| \leq C(\Delta t, \Lambda) \sup_{\nu \in \Lambda} |\mathbb{E} [\tilde{u}^n(x_\nu) - u(x_\nu)]| + C_1(\Delta t, \Lambda)$$

with  $C(\Delta t, \Lambda) = \sup_{\nu \in \Lambda} \sum_{\mu \in \Lambda} |e^{\Delta t}(l_\mu, x_\nu)|$  and  $C_1(\Delta t, \Lambda) = \sup_{\nu \in \Lambda} |e^{\Delta t}(u - \mathcal{I}_\Lambda(u), x_\nu)|$ .

Moreover if  $C(\Delta t, \Lambda) < 1$ , it holds

$$(2.9) \quad \limsup_{n \rightarrow \infty} \sup_{\nu \in \Lambda} |\mathbb{E}[\tilde{u}^n(x_\nu) - u(x_\nu)]| \leq \frac{C_1(\Delta t, \Lambda)}{1 - C(\Delta t, \Lambda)}.$$

The condition  $C(\Delta t, \Lambda) < 1$  implies that in practice  $\Delta t$  should be chosen sufficiently small [49, Theorem 4.2]. Under this condition, the error at interpolation points uniformly converges geometrically up to a threshold term depending on time integration errors for interpolation functions  $l_\nu$  and the interpolation error  $u - \mathcal{I}_\Lambda(u)$ .

Theorem 2.3 provides a convergence result at interpolation points. Below, we provide a corollary to this theorem that provides a convergence result in  $L^\infty(\mathcal{D})$ . This result involves the Lebesgue constants in  $L^\infty$ -norm associated to  $\mathcal{I}_\Lambda$ , defined by

$$(2.10) \quad \mathcal{L}_\Lambda = \sup_{v \in \mathcal{C}^0(\overline{\mathcal{D}})} \frac{\|\mathcal{I}_\Lambda(v)\|_\infty}{\|v\|_\infty},$$

and such that for any  $v \in \mathcal{C}^0(\overline{\mathcal{D}})$ ,

$$(2.11) \quad \|v - \mathcal{I}_\Lambda(v)\|_\infty \leq (1 + \mathcal{L}_\Lambda) \inf_{w \in V_\Lambda} \|v - w\|_\infty.$$

Throughout this article, we adopt the convention that supremum exclude elements with norm 0. We recall also that the  $L^\infty$  Lebesgue constant can be expressed as  $\mathcal{L}_\Lambda = \sup_{x \in \overline{\mathcal{D}}} \sum_{\nu \in \Lambda} |l_\nu(x)|$ .

**Corollary 2.4** (Convergence in  $L^\infty$ ).

Assuming (E2)-(E5), it holds

$$(2.12) \quad \limsup_{n \rightarrow \infty} \|\mathbb{E}[\tilde{u}^n - u]\|_\infty \leq \frac{C_1(\Delta t, \Lambda)}{1 - C(\Delta t, \Lambda)} \mathcal{L}_\Lambda + \|u - \mathcal{I}_\Lambda(u)\|_\infty.$$

*Proof.* By triangular inequality, we have

$$\|\mathbb{E}[\tilde{u}^n - u]\|_\infty \leq \|\mathbb{E}[\tilde{u}^n - \mathcal{I}_\Lambda(u)]\|_\infty + \|\mathcal{I}_\Lambda(u) - u\|_\infty.$$

We can build a continuous function  $w$  such that  $w(x_\nu) = \mathbb{E}[\tilde{u}^n(x_\nu) - u(x_\nu)]$  for all  $\nu \in \Lambda$ , and such that

$$\|w\|_\infty = \sup_{\nu \in \Lambda} |w(x_\nu)| = \sup_{\nu \in \Lambda} |\mathbb{E}[\tilde{u}^n(x_\nu) - u(x_\nu)]|.$$

We have then

$$\|\mathbb{E}[\tilde{u}^n - \mathcal{I}_\Lambda(u)]\|_\infty = \|\mathcal{I}_\Lambda(w)\|_\infty \leq \mathcal{L}_\Lambda \|w\|_\infty.$$

The result follows from the definition of the function  $w$  and Theorem 2.3.  $\square$

**Remark 2.5.** Since for bounded domains  $\mathcal{D}$ , we have

$$\|v\|_2 \leq |\mathcal{D}|^{1/2} \|v\|_\infty,$$

for all  $v$  in  $\mathcal{C}^0(\overline{\mathcal{D}})$ , where  $|\mathcal{D}|$  denotes the Lebesgue measure of  $\mathcal{D}$ , we can deduce the convergence results in  $L^2$  norm from those in  $L^\infty$  norm.

## 2.3 Adaptive Sparse Interpolation

We here present sparse interpolation methods following [21, 22].

### 2.3.1 Sparse Interpolation

For  $1 \leq i \leq d$ , we let  $\{\varphi_k^{(i)}\}_{k \in \mathbb{N}_0}$  be a univariate polynomial basis, where  $\varphi_k^{(i)}(x_i)$  is a polynomial of degree  $k$ . For a multi-index  $\nu = (\nu_1, \dots, \nu_d) \in \mathbb{N}_0^d$ , we introduce the multivariate polynomial

$$\varphi_\nu(x) = \prod_{i=1}^d \varphi_{\nu_i}^{(i)}(x_i).$$

For a subset  $\Lambda \subset \mathbb{N}^d$ , we let  $\mathcal{P}_\Lambda = \text{span}\{\varphi_\nu : \nu \in \Lambda\}$ . A subset  $\Lambda$  is said to be downward closed if

$$\forall \nu \in \Lambda, \mu \leq \nu \Rightarrow \mu \in \Lambda.$$

If  $\Lambda$  is downward closed, then the polynomial space  $\mathcal{P}_\Lambda$  does not depend on the choice of univariate polynomial bases and is such that  $\mathcal{P}_\Lambda = \text{span}\{x^\nu : \nu \in \Lambda\}$ , with  $x^\nu = x_1^{\nu_1} \dots x_d^{\nu_d}$ .

In the case where  $\mathcal{D} = \mathcal{D}_1 \times \dots \times \mathcal{D}_d$ , we can choose for  $\{\varphi_k^{(i)}\}_{k \in \mathbb{N}_0}$  an orthonormal basis in  $L^2(\mathcal{D}_i)$  (i.e. a rescaled and shifted Legendre basis). Then  $\{\varphi_\nu\}_{\nu \in \mathbb{N}_0^d}$  is an orthonormal basis of  $L^2(\mathcal{D})$ . To define a set of points  $\Gamma_\Lambda$  unisolvent for  $\mathcal{P}_\Lambda$ , we can proceed as follows. For each dimension  $1 \leq i \leq d$ , we introduce a sequence of points  $\{z_k^{(i)}\}_{k \in \mathbb{N}_0}$  in  $\overline{\mathcal{D}_i}$  such that for any  $p \geq 0$ ,  $\Gamma_p^{(i)} = \{z_k^{(i)}\}_{k=0}^p$  is unisolvent for  $\mathcal{P}_p = \text{span}\{\varphi_k^{(i)} : 0 \leq k \leq p\}$ , therefore defining an interpolation operator  $\mathcal{I}_p^{(i)}$ . Then we let

$$\Gamma_\Lambda = \{z_\nu = (z_{\nu_1}^{(1)}, \dots, z_{\nu_d}^{(d)}) : \nu \in \Lambda\} \subset \overline{\mathcal{D}}.$$

This construction is interesting for adaptive sparse algorithms since for an increasing sequence of subsets  $\Lambda_n$ , we obtain an increasing sequence of sets  $\Gamma_{\Lambda_n}$ , and the computation of the interpolation on  $\mathcal{P}_{\Lambda_n}$  only requires the evaluation of the function on the new set of points  $\Gamma_{\Lambda_n} \setminus \Gamma_{\Lambda_{n-1}}$ . Also, with such a construction, we have the following property of the Lebesgue constant of  $\mathcal{I}_\Lambda$  in  $L^\infty$ -norm. This result is directly taken from [22, Section 3].

**Proposition 2.6.** *If for each dimension  $1 \leq i \leq d$ , the sequence of points  $\{z_k^{(i)}\}_{k \in \mathbb{N}_0}$  is such that the interpolation operator  $\mathcal{I}_p^{(i)}$  has a Lebesgue constant  $\mathcal{L}_p \leq (p+1)^s$  for some  $s > 0$ , then for any downward closed set  $\Lambda$ , the Lebesgue constant  $\mathcal{L}_\Lambda$  satisfies*

$$(2.13) \quad \mathcal{L}_\Lambda \leq (\#\Lambda)^{s+1}.$$

Leja points or magic points [75] are examples of sequences of points such that the interpolation operators  $\mathcal{I}_p^{(i)}$  have Lebesgue constants not growing too fast with  $p$ . For a given  $\Lambda$  with  $\rho_i := \max_{\nu \in \Lambda} \nu_i$ , it is possible to construct univariate interpolation grids  $\Gamma_{\rho_i}^{(i)}$  with better properties (e.g., Chebychev points), therefore resulting in better properties for the associated interpolation operator  $\mathcal{I}_\Lambda$ . However for Chebychev points, e.g.,  $\rho_i \leq \rho'_i$  does not ensure  $\Gamma_{\rho_i}^{(i)} \subset \Gamma_{\rho'_i}^{(i)}$ . Thus with such univariate grids, an increasing sequence of sets  $\Lambda_n$  will not be associated with an increasing sequence

of sets  $\Gamma_{\Lambda_n}$ , and the evaluations of the function will not be completely recycled in adaptive algorithms. However, for some of the algorithms described in Section 2.4, this is not an issue as evaluations can not be recycled anyway.

Note that for general domains  $\mathcal{D}$  which are not the product of intervals, the above constructions of grids  $\Gamma_\Lambda$  are not viable since it may yield to grids not contained in the domain  $\mathcal{D}$ . For such general domains, magic points obtained through greedy algorithms could be considered.

### 2.3.2 Adaptive Algorithm for Sparse Interpolation

An adaptive sparse interpolation algorithm consists in constructing a sequence of approximations  $(u_n)_{n \geq 1}$  associated with an increasing sequence of downward closed subsets  $(\Lambda_n)_{n \geq 1}$ . According to (2.11), we have to construct a sequence such that the best approximation error and the Lebesgue constant are such that

$$\mathcal{L}_{\Lambda_n} \inf_{w \in \mathcal{P}_{\Lambda_n}} \|u - w\|_\infty \longrightarrow 0 \text{ as } n \rightarrow \infty$$

for obtaining a convergent algorithm. For example, if

$$(2.14) \quad \inf_{w \in \mathcal{P}_{\Lambda_n}} \|u - w\|_\infty = O((\#\Lambda_n)^{-r})$$

holds<sup>[3]</sup> for some  $r > 1$  and if  $\mathcal{L}_{\Lambda_n} = O((\#\Lambda_n)^k)$  for  $k < r$ , then the error  $\|u - u_n\|_\infty = O(n^{-r'})$  tends to zero with an algebraic rate of convergence  $r' = r - k > 0$ . Of course, the challenge is to propose a practical algorithm that constructs a good sequence of sets  $\Lambda_m$ .

We now present the adaptive sparse interpolation algorithm with bulk chasing procedure introduced in [21]. Let  $\theta$  be a fixed bulk chasing parameter in  $(0, 1)$  and let  $\mathcal{E}_\Lambda(v) = \|P_\Lambda(v)\|_2^2$ , where  $P_\Lambda$  is the orthogonal projector over  $\mathcal{P}_\Lambda$  for any subset  $\Lambda \subset \mathbb{N}_0^d$ . At iteration  $n$ , Algorithm 2.2 selects a subset of multi-indices  $N_n$  in the

---

#### Algorithm 2.2 Adaptive Interpolation Algorithm [22]

---

- 1: Set  $\Lambda_1 = \{\mathbf{0}_d\}$  and  $n = 1$ .
  - 2: **while**  $n \leq N$  and  $\varepsilon^{n-1} > \varepsilon$  **do**
  - 3:   Compute  $\mathcal{M}_{\Lambda_n}$ .
  - 4:   Set  $\Lambda_n^* = \Lambda_n \cup \mathcal{M}_{\Lambda_n}$  and compute  $\mathcal{I}_{\Lambda_n^*}(u)$ .
  - 5:   Select  $N_n \subset \mathcal{M}_{\Lambda_n}$  the smallest such that  $\mathcal{E}_{N_n}(\mathcal{I}_{\Lambda_n^*}(u)) \geq \theta \mathcal{E}_{\mathcal{M}_{\Lambda_n}}(\mathcal{I}_{\Lambda_n^*}(u))$
  - 6:   Update  $\Lambda_{n+1} = \Lambda_n \cup N_n$ .
  - 7:   Compute  $u_{n+1} = \mathcal{I}_{\Lambda_{n+1}}(u)$  (this step is not necessary in practice).
  - 8:   Compute  $\varepsilon^n$ .
  - 9:   Update  $n = n + 1$ .
  - 10: **end while**
- 

reduced margin of  $\Lambda_n$  defined by

$$\mathcal{M}_{\Lambda_n} = \{\nu \in \mathbb{N}^d \setminus \Lambda_n : \forall j \text{ s.t. } \nu_j > 0, \nu - e_j \in \Lambda_n\},$$

where  $(e_j)_k = \delta_{kj}$ . The reduced margin is such that for any subset  $S \subset \mathcal{M}_{\Lambda_n}$ ,  $\Lambda_n \cup S$  is downward closed. This ensures that the sequence  $(\Lambda_n)_{n \geq 1}$  generated by

---

<sup>[3]</sup>see e.g. [24] for conditions on  $u$  ensuring such a behavior of the approximation error.

the algorithm is an increasing sequence of downward closed sets. Finally, Algorithm 2.2 is stopped using a criterion based on

$$\varepsilon^n = \frac{\mathcal{E}_{\mathcal{M}_n}(\mathcal{I}_{\Lambda_n^*}(u))}{\mathcal{E}_{\Lambda_n^*}(\mathcal{I}_{\Lambda_n^*}(u))}.$$

## 2.4 Combining Sparse Adaptive Interpolation with Sequential Control Variates Algorithm

We present in this section two ways of combining Algorithm 2.1 and Algorithm 2.2. First we introduce a perturbed version of Algorithm 2.2 and then an adaptive version of Algorithm 2.1. At the end of the section, numerical results will illustrate the behavior of the proposed algorithms.

### 2.4.1 Perturbed Version of Sparse Interpolation

As we do not have access to exact evaluations of the solution  $u$  of (2.1), Algorithm 2.2 can not be used for interpolating  $u$ . So we introduce a perturbed version of this algorithm, where the computation of the exact interpolant  $\mathcal{I}_\Lambda(u)$  is replaced by an approximation denoted  $\tilde{u}_\Lambda$ , which can be computed for example with Algorithm 2.1 stopped for a given tolerance  $\epsilon_{tol}$  or at step  $k$ . This brings the following algorithm.

---

**Algorithm 2.3** Perturbed adaptive sparse interpolation algorithm

---

- 1: Set  $\Lambda_1 = \{\mathbf{0}_d\}$  and  $n = 1$ .
  - 2: **while**  $n \leq N$  and  $\tilde{\varepsilon}^{n-1} > \varepsilon$  **do**
  - 3:   Compute  $\mathcal{M}_{\Lambda_n}$ .
  - 4:   Set  $\Lambda_n^* = \Lambda_n \cup \mathcal{M}_{\Lambda_n}$  and compute  $\tilde{u}_{\Lambda_n^*}$ .
  - 5:   Select  $N_n$  as the smallest subset of  $\mathcal{M}_{\Lambda_n}$  such that  $\mathcal{E}_{N_n}(\tilde{u}_{\Lambda_n^*}) \geq \theta \mathcal{E}_{\mathcal{M}_{\Lambda_n}}(\tilde{u}_{\Lambda_n^*})$
  - 6:   Update  $\Lambda_{n+1} = \Lambda_n \cup N_n$ .
  - 7:   Compute  $\tilde{u}_{\Lambda_{n+1}}$ .
  - 8:   Compute  $\tilde{\varepsilon}^n$ .
  - 9:   Update  $n = n + 1$ .
  - 10: **end while**
- 

### 2.4.2 Adaptive Version of Sequential Control Variate Algorithm

As a second algorithm, we consider the sequential control variates algorithm (Algorithm 2.1) where at step 4, an approximation  $\tilde{e}^k$  of  $e^k$  is obtained by applying the adaptive interpolation algorithm (Algorithm 2.3) to the function  $e_{\Delta t, M}^k$ , which uses Monte-Carlo estimations  $e_{\Delta t, M}^k(x_\nu)$  of  $e^k(x_\nu)$  at interpolation points. At each iteration,  $\tilde{e}^k$  therefore belongs to a different approximation space  $\mathcal{P}_{\Lambda_k}$ . In the numerical section, we will call this algorithm **adaptive Algorithm 2.1**.

### 2.4.3 Numerical Results

In this section, we illustrate the behavior of algorithms previously introduced on different test cases. We consider the simple diffusion equation

$$(2.15) \quad \begin{aligned} -\Delta u(x) &= g(x), & x \in \mathcal{D}, \\ u(x) &= f(x), & x \in \partial\mathcal{D}, \end{aligned}$$

where  $\mathcal{D} = ]-1, 1[^d$ . The source terms and boundary conditions will be specified later for each test case.

The stochastic differential equation associated to (2.15) is the following

$$(2.16) \quad dX_t^x = \sqrt{2}dW_t, \quad X_0^x = x,$$

where  $(W_t)_{t \geq 0}$  is a  $d$ -dimensional Brownian motion.

We use tensorized grids of magic points for the selection of interpolation points evolved in adaptive algorithms.

*Small dimensional test case.* We consider a first test case (TC1) in dimension  $d = 5$ . Here the source term and the boundary conditions in problem (2.15) are chosen such that the solution is given by

$$(TC1) \quad u(x) = x_1^2 + \sin(x_2) + \exp(x_3) + \sin(x_4)(x_5 + 1), \quad x \in \bar{\mathcal{D}}.$$

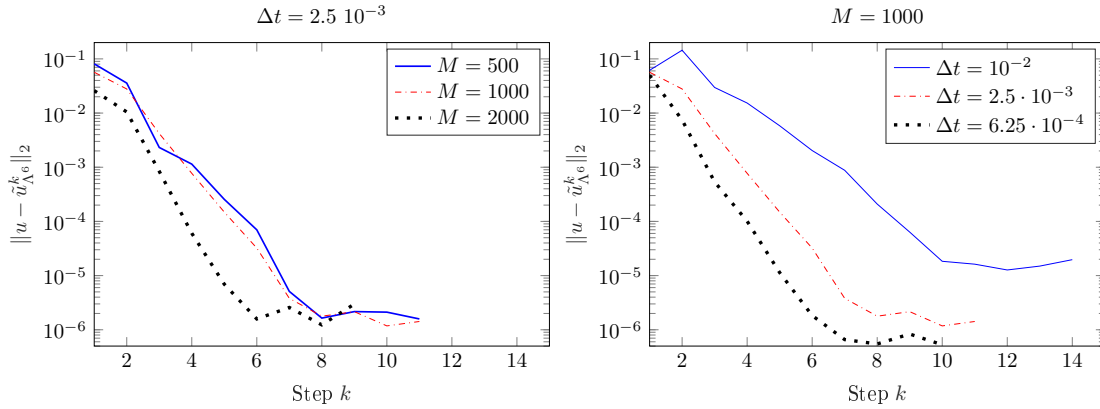


Figure 2.1: (TC1) Algorithm 2.1 for fixed  $\Lambda$  : evolution of  $\|u - \tilde{u}_{\Lambda^6}^k\|$  with respect to  $k$  for various  $M$  (left figure), and various  $\Delta t$  (right figure).

We first test the influence of  $\Delta t$  and  $M$  on the convergence of Algorithm 2.1 when  $\Lambda$  is fixed. In that case,  $\Lambda$  is selected a priori with Algorithm 2.2 using samples of the exact solution  $u$  for (TC1), stopped for  $\varepsilon \in \{10^{-6}, 10^{-8}, 10^{-10}\}$ . In what follows, the notation  $\Lambda^i$  stands for the set obtained for  $\varepsilon = 10^{-i}$ ,  $i \in \{6, 8, 10\}$ . We represent on Figure 2.1 the evolution of the absolute error in  $L^2$ -norm (similar results hold for the  $L^\infty$ -norm) between the approximation and the true solution with respect to step  $k$  for  $\Lambda = \Lambda^6$ . As claimed in Corollary 1, we recover the geometric convergence up to a threshold value that depends on  $\Delta t$ . We also notice faster convergence as  $M$  increases and when  $\Delta t$  decreases. We fix  $M = 1000$  in the next simulations.



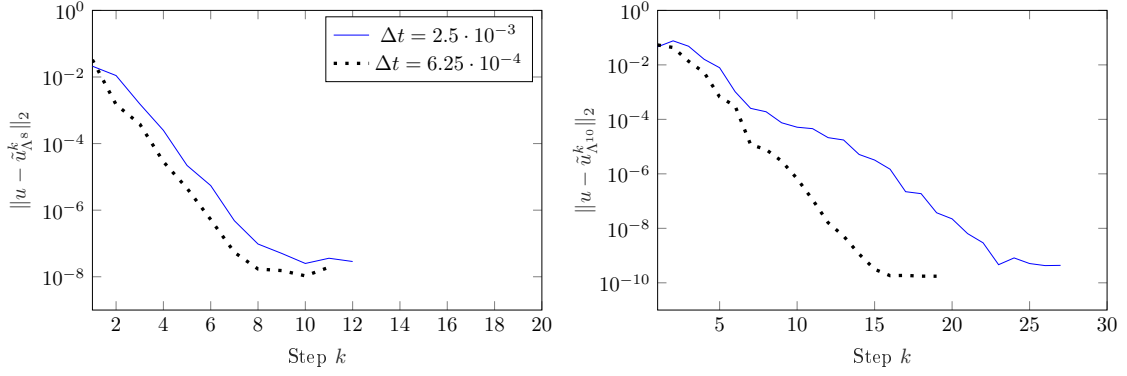


Figure 2.2: (TC1) Algorithm 2.1 for fixed  $\Lambda^i$ : evolution of  $\|u - \tilde{u}_{\Lambda^i}^k\|_2$  with respect to  $k$  for  $i = 8$  (left figure), and  $i = 10$  (right figure).

We study the impact of the choice of  $\Lambda^i$  on the convergence of Algorithm 2.1. Again we observe on Figure 2.2 that the convergence rate gets better as  $\Delta t$  decreases. Moreover as  $\#\Lambda$  increases the threshold value decreases. This is justified by the fact that interpolation error decreases as  $\#\Lambda^i$  increases (see Table 2.1). Nevertheless, we observe that it may also deteriorate the convergence rate if it is chosen too large together with  $\Delta t$  not sufficiently small. Indeed for the same number of iterations  $k = 10$  and the same time-step  $\Delta t = 2.5 \cdot 10^{-3}$ , we have an approximate absolute error equal to  $10^{-7}$  for  $\Lambda^8$  against  $10^{-4}$  for  $\Lambda^{10}$ .

$\Lambda_n$	$\#\Lambda_n$	$\varepsilon^n$	$\ u - u_n\ _2$	$\ u - u_n\ _\infty$
$\Lambda^6$	1	6.183372e-01	1.261601e+00	4.213566e+00
	10	2.792486e-02	1.204421e-01	3.602629e-01
	20	2.178450e-05	9.394419e-04	3.393999e-03
	26	9.632815e-07	4.270457e-06	1.585129e-05
	30	9.699704e-08	2.447475e-06	8.316435e-06
$\Lambda^8$	33	4.114730e-09	2.189518e-08	9.880306e-08
	40	1.936050e-10	6.135776e-10	1.739848e-09
$\Lambda^{10}$	41	1.008412e-11	9.535433e-11	4.781375e-10
	50	1.900248e-14	1.004230e-13	4.223288e-13
	55	7.453467e-15	2.905404e-14	1.254552e-13

Table 2.1: Algorithm 2.2 computed on the exact solution of (TC1): evolution of  $\#\Lambda_n$ , error criterion  $\varepsilon^n$  and interpolation errors in norms  $L^2$  and  $L^\infty$  at each step  $n$ .

We present now the behavior of Algorithm 2.3. Simulations are performed with a bulk-chasing parameter  $\theta = 0.5$ . At each step  $n$  of Algorithm 2.3, we use Algorithm 2.1 with  $(\Delta t, M) = (10^{-4}, 1000)$ , stopped when a stagnation is detected. As shown on the left plot of Figure 2.3, for  $\#\Lambda_n = 55$  we reach approximately a precision of  $10^{-14}$  as for Algorithm 2.2 performed on the exact solution (see Table 2.1). According to the right plot of Figure 2.3, we also observe that the enrichment procedure behaves similarly for both algorithms ( $\tilde{\varepsilon}^n$  and  $\varepsilon^n$  are almost the same). Here using the approximation provided by Algorithm 2.1 has a low impact on the behavior of

Algorithm 2.2.

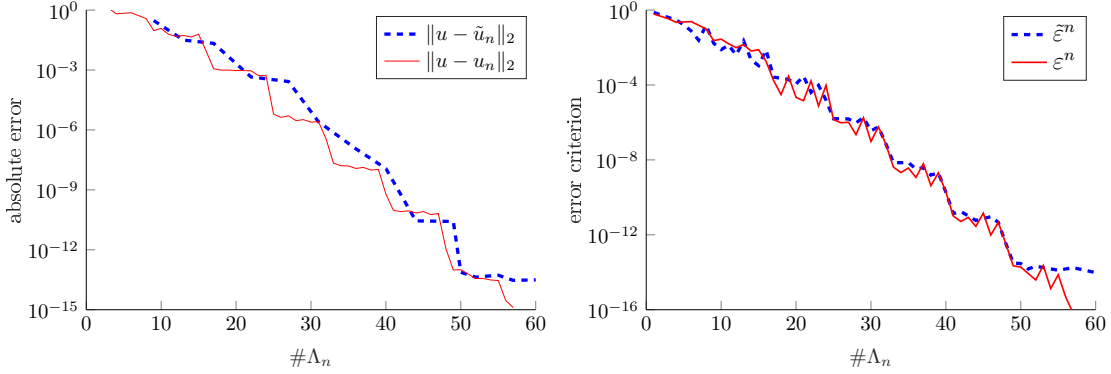


Figure 2.3: (TC1) Comparison of Algorithm 2.2 applied to exact solution and Algorithm 2.3 : (left) absolute error in  $L^2$ -norm (right) evolution of  $\varepsilon^n$  and  $\tilde{\varepsilon}^n$  with respect to  $\#\Lambda_n$ .

We present then results provided with the adaptive Algorithm 2.1. The parameters chosen for the adaptive interpolation are  $\varepsilon = 5 \cdot 10^{-2}$ ,  $\theta = 0.5$ .  $K = 30$  ensures the stopping of Algorithm 2.1. As illustrated by Figure 2.4, we recover globally the same behavior as for Algorithm 2.1 without adaptive interpolation. Indeed as  $k$  increases, both absolute errors in  $L^2$ -norm and  $L^\infty$ -norm decrease and then stagnate. Again, we notice the influence of  $\Delta t$  on the stagnation level. Nevertheless, the convergence rates are deteriorated and the algorithm provides less accurate approximations than Algorithm 2.3. This might be due to the sparse adaptive interpolation procedure, which uses here pointwise evaluations based on Monte-Carlo estimates, unlike Algorithm 2.3 which relies on pointwise evaluations resulting from Algorithm 2.1 stopping for a given tolerance.

Finally in Table 2.2, we compare the algorithmic complexity of these algorithms to reach a precision of  $3 \cdot 10^{-5}$  for  $(\Delta t, M) = (10^{-4}, 1000)$ . For adaptive Algorithm 2.1,  $\Lambda_k$  refers to the set of multi-indices considered at step  $k$  of Algorithm 2.1. For Algorithm 2.3,  $N_n$  stands for the number of iteration required by Algorithm 2.1 to reach tolerance  $\varepsilon_{tol}$  at step  $n$ . Finally, Algorithm 2.1 is run with full-grid  $\Lambda = \Lambda_{max}$  where  $\Lambda_{max} = \{\nu \in \mathbb{N}^d : \nu_i \leq 10\}$  is the set of multi-indices allowing to reach the machine precision. In this case,  $N$  stands for the number of steps for this algorithm to converge.

	Adaptive Algorithm 2.1	Algorithm 2.3	Full-grid Algorithm 2.1
Th. Complexity	$M(\Delta t)^{-1}(\sum_k \#\Lambda_k)$	$M(\Delta t)^{-1}(\sum_n \#\Lambda_n N_n)$	$M(\Delta t)^{-1} \#\Lambda_{max} N$
Est. Complexity	$4 \cdot 10^9$ operations	$16 \cdot 10^9$ operations	$10^{12} N$ operations

Table 2.2: (TC1) Comparison of the algorithmic complexity to reach the precision  $3 \cdot 10^{-5}$ , with  $(\Delta t, M) = (10^{-4}, 1000)$ .

We observe that both the adaptive version of Algorithm 2.1 and Algorithm 2.3 have a similar complexity, which is better than for the full-grid version of Algorithm 2.1.

Moreover, we observed that while adaptive version of Algorithm 2.1 stagnates at a precision of  $3 \cdot 10^{-5}$ , Algorithm 2.3, with the same parameters  $\Delta t$  and  $M$ , converges almost up to the machine precision. This is why the high-dimensional test cases will be run only with Algorithm 2.3.

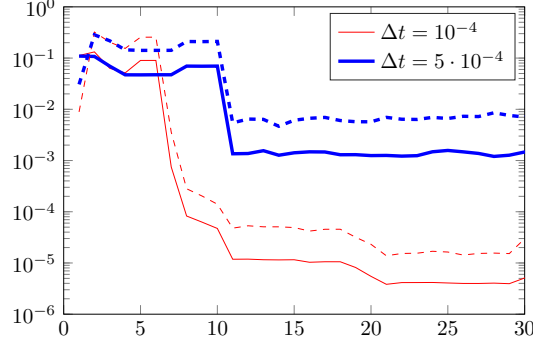


Figure 2.4: (TC1) Adaptive Algorithm 2.1: evolution of  $\|u - u_{\Lambda_k}^k\|_2$  (continuous line) and  $\|u - u_{\Lambda_k}^k\|_\infty$  (dashed line) with respect to step  $k$  and  $\Delta t$ .

*Higher-dimensional test cases.* Now, we consider two other test cases noted respectively (TC2) and (TC3) in higher dimension.

(TC2) As second test case in dimension  $d = 10$ , we define (2.15) such that its solution is the Henon-Heiles potential

$$u(x) = \frac{1}{2} \sum_{i=1}^d x_i^2 + 0.2 \sum_{i=1}^{d-1} (x_i x_{i+1}^2 - x_i^3) + 2.5 \cdot 10^{-3} \sum_{i=1}^{d-1} (x_i^2 + x_{i+1}^2)^2, \quad x \in \overline{\mathcal{D}}.$$

We set  $(\Delta t, M) = (10^{-4}, 1000)$  and  $K = 30$  for Algorithm 2.1.

(TC3) We also consider the problem (2.15) whose exact solution is a sum of non-polynomial functions, like (TC1) but now in dimension  $d = 20$ , given by

$$u(x) = x_1^2 + \sin(x_{12}) + \exp(x_5) + \sin(x_{15})(x_8 + 1).$$

Here, the Monte-Carlo simulations are performed for  $(\Delta t, M) = (10^{-4}, 1000)$  and  $K = 30$ .

Since for both test cases the exact solution is known, we propose to compare the behavior of Algorithm 2.3 and Algorithm 2.2. Again, the approximations  $\tilde{u}_n$ , at each step  $n$  of Algorithm 2.3, are provided by Algorithm 2.1 stopped when a stagnation is detected. In both cases, the parameters for Algorithm 2.3 are set to  $\theta = 0.5$  and  $\varepsilon = 10^{-15}$ .

In Table 2.3 and Table 2.4, we summarize the results associated to the exact and perturbed sparse adaptive algorithms for (TC2) and (TC3) respectively. We observe that Algorithm 2.3 performs well in comparison to Algorithm 2.2, for (TC2). Indeed, we get an approximation with a precision below the prescribed value  $\varepsilon$  for both algorithms.

$\#\Lambda_n$	$\varepsilon^n$	$\ u - u_n\ _\infty$	$\ u - u_n\ _2$	$\#\Lambda_n$	$\tilde{\varepsilon}^n$	$\ u - \tilde{u}_{\Lambda_n}\ _\infty$	$\ u - \tilde{u}_{\Lambda_n}\ _2$
1	4.0523e-01	3.0151e+00	1.2094e+00	1	3.9118e-01	8.3958e-01	6.9168e-01
17	1.6243e-01	1.8876e+00	5.9579e-01	17	1.6259e-01	5.2498e-01	3.4420e-01
36	5.4494e-02	7.0219e-01	2.0016e-01	36	5.4699e-02	1.9209e-01	1.2594e-01
46	1.2767e-02	1.6715e-01	4.9736e-02	46	1.2806e-02	4.6904e-02	2.8524e-02
53	9.6987e-04	2.9343e-02	4.8820e-03	53	1.0350e-03	7.8754e-03	2.8960e-03
60	7.6753e-04	1.5475e-02	4.1979e-03	61	7.0354e-04	3.0365e-03	1.7610e-03
71	3.2532e-04	8.4575e-03	2.1450e-03	71	3.1998e-04	2.3486e-03	1.2395e-03
77	1.7434e-16	3.9968e-15	1.5784e-15	77	7.3621e-16	6.2172e-15	1.2874e-15

Table 2.3: (TC2) Comparison of Algorithm 2.2 (first four columns) and Algorithm 2.3 (last four columns).

Similar observation holds for (TC3) in Table 2.4 and this despite the fact that the test case involves higher dimensional problem.

$\#\Lambda_n$	$\varepsilon^n$	$\ u - u_n\ _\infty$	$\ u - u_n\ _2$	$\#\Lambda_n$	$\tilde{\varepsilon}^n$	$\ u - \tilde{u}_{\Lambda_n}\ _\infty$	$\ u - \tilde{u}_{\Lambda_n}\ _2$
1	7.0155e-01	3.9361e+00	1.2194e+00	1	5.5582e-01	7.2832e-01	7.0771e-01
6	1.4749e-01	2.2705e+00	5.4886e-01	6	7.4253e-02	2.7579e-01	5.1539e-01
11	2.1902e-02	2.8669e-01	1.0829e-01	11	1.4929e-02	4.4614e-02	4.1973e-02
15	7.6086e-03	1.6425e-01	4.7394e-02	15	1.2916e-02	1.5567e-02	2.5650e-02
20	2.2275e-04	2.7715e-03	7.2230e-04	20	3.4446e-04	5.6927e-04	5.3597e-04
24	1.4581e-05	1.5564e-04	7.5314e-05	24	1.6036e-05	2.5952e-05	3.0835e-05
30	1.8263e-06	8.0838e-06	2.1924e-06	30	9.0141e-07	2.8808e-06	1.9451e-06
35	3.9219e-09	8.9815e-08	2.4651e-08	35	8.1962e-09	2.1927e-08	1.5127e-08
40	1.7933e-10	2.0152e-09	6.9097e-10	40	1.6755e-10	2.8455e-10	2.6952e-10
45	5.0775e-12	2.4783e-10	4.1600e-11	45	1.4627e-11	3.3188e-11	1.7911e-11
49	1.7722e-14	4.6274e-13	8.5980e-14	49	1.7938e-14	8.6362e-14	5.0992e-14
54	3.9609e-15	2.2681e-13	3.1952e-14	54	3.2195e-15	4.8142e-14	2.6617e-14
56	4.5746e-16	8.4376e-15	3.0438e-15	56	8.2539e-16	8.4376e-15	6.3039e-15

Table 2.4: (TC3) Comparison of Algorithm 2.2 (first four columns) and Algorithm 2.3 (last four columns).

## 2.5 Conclusion

In this paper we have introduced a probabilistic approach to approximate the solution of high-dimensional elliptic PDEs. This approach relies on adaptive sparse polynomial interpolation using pointwise evaluations of the solution estimated using a Monte-Carlo method with control variates.

Especially, we have proposed and compared different algorithms. First we proposed Algorithm 2.1 which combines the sequential algorithm proposed in [48] and sparse interpolation. For the non-adaptive version of this algorithm we recover the convergence up to a threshold as the original sequential algorithm [49]. Nevertheless it remains limited to small-dimensional test cases, since its algorithmic complexity remains high. Hence, for practical use, the adaptive Algorithm 2.1 should be preferred. Adaptive Algorithm 2.1 converges but it does not allow to reach low precision with reasonable number of Monte-Carlo samples or time-step in the Euler-Maruyama scheme. Secondly, we proposed Algorithm 2.3. It is a perturbed sparse adaptive interpolation algorithm relying on inexact pointwise evaluations of the function to

approximate. Numerical experiments have shown that the perturbed algorithm (Algorithm 2.3) performs well in comparison to the ideal one (Algorithm 2.2) and better than the adapted Algorithm 2.1 with a similar algorithmic complexity. Here, since only heuristic tools have been provided to justify the convergence of this algorithm, the proof of its convergence, under assumptions on the class of functions to be approximated, should be addressed in a future work.

# Chapter 3

## A PAC Algorithm in Relative Precision for Bandit Problem with Costly Sampling

This chapter introduces our second contribution. It consists in a discrete optimization algorithm whose output is probably quasi-optimal in relative precision, with controlled probability and relative quasi-optimality.

### Contents

3.1	Introduction . . . . .	62
3.2	Monte-Carlo Estimate with Guaranteed Relative Precision	63
3.2.1	Monte-Carlo Estimate . . . . .	64
3.2.2	Complexity Analysis . . . . .	66
3.3	Optimization Algorithms with Guaranteed Relative Precision	66
3.3.1	Non-Adaptive Algorithm . . . . .	67
3.3.2	Adaptive Algorithm . . . . .	69
3.4	Numerical Results . . . . .	71
3.A	Intermediate Results . . . . .	75
3.B	Proof of Proposition 3.4 . . . . .	77

We present in this chapter a contribution that has been submitted as [12] in Mathematical Methods of Operations Research and is currently under review.

This contribution is about discrete optimization for functions defined using a expectation. The probably approximately correct (PAC) algorithm in relative precision presented and analysed in this chapter will then be used in Chapter 5 as a tool to select a quasi-optimal snapshot to enrich the reduced space.

---

**Title:** A PAC algorithm in relative precision for bandit problem with costly sampling

**Authors:** Marie Billaud-Friess ♣, Arthur Macherey ♣♠, Anthony Nouy ♣ and Clémentine Prieur ♠.

♣ Centrale Nantes, LMJL, UMR CNRS 6629, 1 rue de la Noë, 44321 Nantes

♠ Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP\*, LJK, 38000 Grenoble, France

\*Institute of Engineering Univ. Grenoble Alpes

**Abstract:** This paper considers the problem of maximizing an expectation function over a finite set, or finite-arm bandit problem. We first propose a naive stochastic bandit algorithm for obtaining a probably approximately correct (PAC) solution to this discrete optimization problem in relative precision, that is a solution which solves the optimization problem up to a relative error smaller than a prescribed tolerance, with high probability. We also propose an adaptive stochastic bandit algorithm which provides a PAC-solution with the same guarantees. The adaptive algorithm outperforms the mean complexity of the naive algorithm in terms of number of generated samples and is particularly well suited for applications with high sampling cost.

---

## 3.1 Introduction

We consider an optimization problem

$$(3.1) \quad \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)],$$

where  $\mathbb{E}[Z(\xi)]$  is the expectation of a random variable  $Z(\xi)$ , and where we assume that the set  $\Xi$  is finite. Such a problem is encountered in different fields such as reinforcement learning [98] or robust optimization [10].

To solve (3.1), classical optimization methods include random search algorithms [51, 102], stochastic approximation methods [33, 83] and bandit algorithms [73, 2, 3, 43]. In this paper, we focus on unstructured stochastic bandit problems with a finite number of arms where "arms" stands for "random variables" and corresponds here to the  $Z(\xi)$ ,  $\xi \in \Xi$  (see, e.g., [73, Section 4]). Stochastic means that the only way to learn about the probability distribution of arms  $Z(\xi)$ ,  $\xi \in \Xi$  is to generate i.i.d. samples from it. Unstructured means that knowledge about the probability distribution of one arm  $Z(\xi)$  does not restrict the range of possibilities for other arms  $Z(\xi')$ ,  $\xi' \neq \xi$ .

Additionally, we suppose here it is numerically costly to sample the random variables  $Z(\xi)$ ,  $\xi \in \Xi$ . Our aim is thus to solve (3.1) by sampling as few as possible the random variables  $Z(\xi)$ ,  $\xi \in \Xi$ . However, it is not feasible to solve (3.1) almost surely using only a finite number of samples from the random variables  $Z(\xi)$ ,  $\xi \in \Xi$ . Thus, it is relevant to adopt a probably approximately correct (PAC) approach (see e.g. [37]). For a precision  $\tau_{abs}$  and a probability  $\lambda \in (0, 1)$ , a  $(\tau_{abs}, \lambda)$ -PAC algorithm returns  $\hat{\xi}$  such that

$$(3.2) \quad \mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \tau_{abs} \right) \geq 1 - \lambda, \quad \xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)].$$

Until recently, one of the main focus of bandit algorithms was the best arm (random variable) identification [2], through the use of Successive Rejects algorithm or Up-

per Confidence Bounds algorithms. Such algorithms are  $(0, \lambda)$ -PAC algorithms, as stated in [37]. Racing algorithms [3] were designed to solve the best arm identification problem too and are mainly analyzed in a finite budget setting, which consists in fixing a maximum number of samples that can be used. While trying to identify the best arm, bandit algorithms also aim at minimizing the regret [1, 17, 43]. More recently, other focuses have emerged, such as the identification of the subset of  $\Xi$  containing the  $m$  best arms [66, 69] or the identification of "good arms" (also known as thresholding bandit problem) that are random variables whose expectation is greater or equal to a given threshold [67, 100, 81, 74].

The  $(\tau_{abs}, \lambda)$ -PAC algorithms mentioned above measure the error in absolute precision. However, without knowing  $\mathbb{E}[Z(\xi^*)]$ , providing in advance a relevant value for  $\tau_{abs}$  is not an easy task. In this work, we rather consider  $(\tau, \lambda)$ -PAC algorithms in relative precision that return  $\hat{\xi} \in \Xi$  such that

$$(3.3) \quad \mathbb{P} \left( \left| \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \right| \leq \tau |\mathbb{E}[Z(\xi^*)]| \right) \geq 1 - \lambda,$$

where  $\tau$  and  $\lambda$  are set in advance in  $(0, 1)$ . We introduce two algorithms that yield a solution  $\hat{\xi}$  satisfying (3.3). The first algorithm builds an estimate precise enough for each expectation  $\mathbb{E}[Z(\xi)]$ . This naive approach drives a majority of the budget on the random variables with the lowest expectations in absolute value. In order to avoid this drawback and thus to reduce the number of samples required to reach the prescribed relative precision, we propose a second algorithm which adaptively samples random variables exploiting confidence intervals obtained from an empirical Bernstein concentration inequality.

The outline of the paper is as follows. In section 3.2, we present a Monte-Carlo estimate for the expectation of a single random variable that has been proposed in [78]. It provides an estimation of the expectation with guaranteed relative precision, with high probability. In section 3.3, we introduce two new algorithms that rely on these Monte-Carlo estimates and yield a solution to (3.3). Then, we study numerically the performance of our algorithms and compare them to algorithms from the literature, possibly adapted to solve (3.3).

## 3.2 Monte-Carlo Estimate with Guaranteed Relative Precision

In what follows, we consider a random variable  $Z$  defined on probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ . We denote by  $\bar{Z}_m$  the empirical mean of  $Z$  and by  $\bar{V}_m$  its empirical variance, respectively defined by

$$\bar{Z}_m = \frac{1}{m} \sum_{i=1}^m Z_i \quad \text{and} \quad \bar{V}_m = \frac{1}{m} \sum_{i=1}^m (Z_i - \bar{Z}_m)^2,$$

where  $(Z_i)_{i \geq 1}$  is a sequence of i.i.d. copies of  $Z$ . The aim is to provide an estimate  $\hat{\mathbb{E}}[Z]$  of  $\mathbb{E}[Z]$  which satisfies

$$(3.4) \quad \mathbb{P} \left( |\hat{\mathbb{E}}[Z] - \mathbb{E}[Z]| \leq \epsilon |\mathbb{E}[Z]| \right) \geq 1 - \delta,$$



with  $(\epsilon, \delta) \in (0, 1)^2$  given a priori. For that, we will rely on Theorem 3.1 hereafter.

**Theorem 3.1.** *If  $Z$  takes its values in a bounded interval  $[a, b]$ , for any  $m \in \mathbb{N}$  and  $x \in (0, 1)$ , we have*

$$(3.5) \quad \mathbb{P} \left( \left| \bar{Z}_m - \mathbb{E}[Z] \right| \leq \sqrt{\frac{2\bar{V}_m \log(3/x)}{m}} + \frac{3(b-a) \log(3/x)}{m} \right) \geq 1 - x.$$

*Proof.* We simply apply [1, Theorem 1] to  $Z - a$  which is a positive random variable whose values are lower than  $b - a$ .  $\square$

Based on Theorem 3.1, several estimates for  $\mathbb{E}(Z)$  have been proposed in [78, 79]. We focus in this paper on the estimate introduced in [78, Equation (3.7)].

### 3.2.1 Monte-Carlo Estimate

Considering a sequence  $(d_m)_{m \geq 1}$  in  $(0, 1)$ , we introduce the sequence  $(c_m)_{m \geq 1}$  defined, for all  $m \geq 1$ , by

$$(3.6) \quad c_m = \sqrt{\frac{2\bar{V}_m \log(3/d_m)}{m}} + \frac{3(b-a) \log(3/d_m)}{m}.$$

Using Theorem 3.1, we see that  $c_m$  stands for the half-length of a confidence interval of level  $1 - d_m$  for  $\mathbb{E}[Z]$ , i.e.

$$(3.7) \quad \mathbb{P}(|\bar{Z}_m - \mathbb{E}[Z]| \leq c_m) \geq 1 - d_m.$$

Let  $M$  be an integer-valued random variable on  $(\Omega, \mathcal{F}, \mathbb{P})$ , such that

$$(3.8) \quad c_M \leq \epsilon |\bar{Z}_M|,$$

with  $\epsilon \in (0, 1)$ . Then, we define the following estimate

$$(3.9) \quad \hat{\mathbb{E}}_M[Z] = \bar{Z}_M - \epsilon \operatorname{sign}(\bar{Z}_M) c_M.$$

**Proposition 3.2.** *Let  $\epsilon, \delta \in (0, 1)$ . Assume that  $Z$  takes its values in a bounded interval  $[a, b]$  and that  $(d_m)_{m \geq 1}$  satisfies*

$$(3.10) \quad \sum_{m \geq 1} d_m \leq \delta.$$

*Then the estimate  $\hat{\mathbb{E}}_M[Z]$  defined by (3.9), with  $M$  satisfying (3.8), is such that*

$$(3.11) \quad \mathbb{P} \left( \left| \hat{\mathbb{E}}_M[Z] - \mathbb{E}[Z] \right| \leq \epsilon |\mathbb{E}[Z]| \right) \geq 1 - \delta.$$

*Proof.* We have

$$\begin{aligned}
 \mathbb{P}(|\bar{Z}_M - \mathbb{E}[Z]| \leq c_M) &\geq \mathbb{P}\left(\bigcap_{m \geq 1} \{|\bar{Z}_m - \mathbb{E}[Z]| \leq c_m\}\right) \\
 (3.12) \qquad \qquad \qquad &\geq 1 - \sum_{m=1}^{+\infty} \mathbb{P}(|\bar{Z}_m - \mathbb{E}[Z]| > c_m).
 \end{aligned}$$

Then using (3.7) and (3.10), we deduce that

$$(3.13) \qquad \mathbb{P}(|\bar{Z}_M - \mathbb{E}[Z]| \leq c_M) \geq 1 - \sum_{m=1}^{+\infty} d_m \geq 1 - \delta.$$

It remains to prove that  $\{|\bar{Z}_M - \mathbb{E}[Z]| \leq c_M\}$  implies  $\{|\hat{\mathbb{E}}_M[Z] - \mathbb{E}[Z]| \leq \epsilon|\mathbb{E}[Z]|\}$ . In the rest of the proof, we assume that  $|\bar{Z}_M - \mathbb{E}[Z]| \leq c_M$  holds. Let us recall that  $c_M \leq \epsilon|\bar{Z}_M|$ . Then, since  $\epsilon < 1$ , we have

$$|\bar{Z}_M - \mathbb{E}[Z]| \leq c_M \leq \epsilon|\bar{Z}_M| < |\bar{Z}_M|,$$

which implies that  $\mathbb{E}[Z]$ ,  $\bar{Z}_M$  and  $\hat{\mathbb{E}}_M[Z]$  have the same sign. Therefore,

$$|\hat{\mathbb{E}}_M[Z] - \mathbb{E}[Z]| = \left| |\bar{Z}_M| - \epsilon c_M - |\mathbb{E}[Z]| \right|.$$

It suffices to consider the case  $\mathbb{E}[Z] > 0$  and we have

$$\mathbb{E}[Z] \geq \bar{Z}_M - c_M \geq \frac{1-\epsilon}{\epsilon} c_M.$$

Therefore

$$\hat{\mathbb{E}}_M[Z] - \mathbb{E}[Z] = \bar{Z}_M - \mathbb{E}[Z] - \epsilon c_M \leq (1-\epsilon)c_M \leq \epsilon\mathbb{E}[Z].$$

Also

$$\mathbb{E}[Z] \leq \bar{Z}_M + c_M \leq (1+\epsilon)\bar{Z}_M,$$

and

$$\hat{\mathbb{E}}_M[Z] - \mathbb{E}[Z] \geq (1-\epsilon^2)\bar{Z}_M - \mathbb{E}[Z] \geq (1-\epsilon)\mathbb{E}[Z] - \mathbb{E}[Z] = -\epsilon\mathbb{E}[Z],$$

which concludes the proof.  $\square$

In practice, the computation of the estimate given by (3.9) requires a particular choice for the random variable  $M$  and for the sequence  $(d_m)_{m \geq 1}$ . A natural choice for  $M$  which satisfies (3.8) is

$$(3.14) \qquad M = \min \left\{ m \in \mathbb{N}^* : c_m \leq \epsilon|\bar{Z}_m| \right\}.$$

If the sequence  $(d_m)_{m \geq 1}$  is such that

$$(3.15) \qquad \log(3/d_m)/m \xrightarrow{m \rightarrow +\infty} 0,$$

we have that  $c_m$  converges to 0 almost surely. Moreover if  $\mathbb{E}[Z] \neq 0$ , it is sufficient to ensure that  $M < +\infty$  almost surely.

**Remark 3.3.** When choosing  $M$  as in (3.14), the estimate defined by (3.9) is the one proposed in [78, equation (3.7)]. A variant of this estimate can be found in [79].

### 3.2.2 Complexity Analysis

In this section, we state a complexity result. Following [78], we focus on a particular sequence  $(d_m)_{m \geq 1}$  defined by

$$(3.16) \quad d_m = \delta c m^{-p}, \quad c = \frac{p-1}{p},$$

which satisfies (3.10) and (3.15), for any  $p > 1$ . The following result extends the result of [78, Theorem 2] stated for random variables  $Z$  with range in  $[0, 1]$ .

**Proposition 3.4.** *Let  $0 < \delta \leq 3/4$  and let  $Z$  be a random variable taking values in a bounded interval  $[a, b]$ , with expectation  $\mu = \mathbb{E}[Z]$  and variance  $\sigma^2 = \mathbb{V}[Z]$ . If  $\mu \neq 0$  and  $(d_m)_{m \geq 1}$  satisfies (3.16), then  $M$  defined by (3.14) satisfies  $M < +\infty$  almost surely and*

$$(3.17) \quad \mathbb{P} \left( M > \left\lceil \frac{2}{\nu} \left( p \log \left( \frac{2p}{\nu} \right) + \log \left( \frac{3}{c\delta} \right) \right) \right\rceil \right) \leq 4\delta/3,$$

where  $\lceil \cdot \rceil$  denotes the ceil function and

$$\nu = \min \left( \frac{\max(\sigma^2, \epsilon^2 \mu^2)}{(b-a)^2}, \frac{\epsilon^2 \mu^2}{(1+\epsilon)^2 \max(\sigma^2, \epsilon^2 \mu^2) \gamma} \right),$$

with  $\gamma = (\sqrt{2 + 2\sqrt{2}} + 2/3 + 3)^2$ . Moreover,

$$\mathbb{E}(M) \leq \left\lceil \frac{2}{\nu} \left( p \log \left( \frac{2p}{\nu} \right) + \log \left( \frac{3}{c\delta} \right) \right) \right\rceil + 4\delta/3.$$

*Proof.* See Appendix. □

**Remark 3.5.** *The result from Proposition 3.4 helps in understanding the influence of parameters  $(\epsilon, \delta)$  appearing in (3.4) on  $M$ . Indeed, we deduce from this result that for  $\delta < 1/2$ ,*

$$\mathbb{E}(M) \lesssim \nu^{-1} \log(\nu^{-1}) + (\nu^{-1} + 1) \log(\delta^{-1}).$$

*We first observe a weak impact of  $\delta$  on the average complexity. When  $\epsilon \rightarrow 0$ , we have  $\nu \sim \epsilon^2 \frac{\mu^2}{\sigma^2 \gamma}$ . Then for fixed  $\delta$  and  $\epsilon \rightarrow 0$ , the bound for  $\mathbb{E}(M)$  is in  $O(\epsilon^{-2} \frac{\sigma^2}{\mu^2})$ . As expected, the relative precision  $\epsilon$  has a much stronger impact on the average complexity.*

## 3.3 Optimization Algorithms with Guaranteed Relative Precision

In this section we consider a finite collection of bounded random variables  $Z(\xi)$  on  $(\Omega, \mathcal{F}, \mathbb{P})$ , indexed by  $\xi \in \Xi$ , and such that  $\mathbb{E}[Z(\xi)] \neq 0$ . Each random variable  $Z(\xi)$  takes its values in a bounded interval  $[a(\xi), b(\xi)]$ , which is assumed to be known.

We denote by  $\overline{Z(\xi)}_m$  the empirical mean of  $Z(\xi)$  and  $\overline{V(\xi)}_m$  its empirical variance, respectively defined by

$$(3.18) \quad \overline{Z(\xi)}_m = \frac{1}{m} \sum_{i=1}^m Z(\xi)_i \quad \text{and} \quad \overline{V(\xi)}_m = \frac{1}{m} \sum_{i=1}^m \left( Z(\xi)_i - \overline{Z(\xi)}_m \right)^2,$$

where  $\{(Z(\xi)_i)_{i \geq 1} : \xi \in \Xi\}$  are independent i.i.d. copies of  $\{Z(\xi) : \xi \in \Xi\}$ . We also introduce  $\#\Xi$  different sequences

$$c_{\xi,m} = \sqrt{\frac{2\overline{V(\xi)}_m \log(3/d_m)}{m}} + \frac{3(b(\xi) - a(\xi)) \log(3/d_m)}{m},$$

where  $(d_m)_{m \geq 1}$  is a positive sequence, independent from  $\xi$ , such that  $\sum_{m \geq 1} d_m \leq \delta$ . Taking  $\epsilon$  in  $(0, 1)$ , for each  $\xi$  in  $\Xi$ , we define, as in (3.14),

$$(3.19) \quad m(\xi) = \min \left\{ m \in \mathbb{N}^* : c_{\xi,m} \leq \epsilon |\overline{Z(\xi)}_m| \right\}.$$

Then defining  $s(\xi) := \text{sign}(\overline{Z(\xi)}_{m(\xi)})$ , we propose the following estimate for  $\mathbb{E}[Z(\xi)]$ :

$$(3.20) \quad \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)] = \overline{Z(\xi)}_{m(\xi)} - \epsilon s(\xi) c_{\xi,m(\xi)}.$$

These notation being introduced, we propose below two algorithms returning  $\hat{\xi}$  in  $\Xi$  such that

$$(3.21) \quad \mathbb{P} \left( \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \tau |\mathbb{E}[Z(\xi^*)]| \right) \geq 1 - \lambda, \quad \xi^* \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z(\xi)],$$

for given  $(\tau, \lambda)$  in  $(0, 1)^2$ .

### 3.3.1 Non-Adaptive Algorithm

We first propose a non-adaptive algorithm that provides a parameter  $\hat{\xi}$  satisfying (3.21), by selecting the maximizer of independent estimates  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  of  $\mathbb{E}[Z(\xi)]$  over  $\Xi$ .

---

#### Algorithm 3.1 Non-Adaptive PAC Algorithm in Relative Precision

---

**Require:**  $\tau, \lambda, \{Z(\xi)\}_{\xi \in \Xi}$ .

**Ensure:**  $\hat{\xi}$

- 1: Set  $\epsilon = \frac{\tau}{2+\tau}$  and  $\delta = \lambda/\#\Xi$ .
- 2: **for all**  $\xi \in \Xi$  **do**
- 3:   Build an estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  of  $\mathbb{E}[Z(\xi)]$  using (3.20) with  $\epsilon$  and  $\delta$  as above.
- 4: **end for**
- 5: Select  $\hat{\xi}$  such that

$$\hat{\xi} \in \arg \max_{\xi \in \Xi} \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)].$$


---

**Proposition 3.6.** *Let  $(\tau, \lambda) \in (0, 1)^2$ . We assume that, for all  $\xi \in \Xi$ ,  $Z(\xi)$  is a bounded random variable with  $\mathbb{E}[Z(\xi)] \neq 0$ . Moreover we assume that the sequence  $(d_m)_{m \geq 1}$  is such that*

$$(3.22) \quad \sum_{m=1}^{+\infty} d_m \leq \frac{\lambda}{\#\Xi} := \delta \quad \text{and} \quad \log(3/d_m)/m \xrightarrow{m \rightarrow +\infty} 0.$$

*Then, for all  $\xi$  in  $\Xi$ , the estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  is well defined and satisfies*

$$(3.23) \quad \mathbb{P}\left(\left|\mathbb{E}[Z(\xi)] - \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]\right| \leq \epsilon |\mathbb{E}[Z(\xi)]|\right) \geq 1 - \delta,$$

*with  $\epsilon = \frac{\tau}{2+\tau}$ . Moreover, the value  $\hat{\xi}$  returned by Algorithm 3.1 satisfies (3.21).*

*Proof.* The assumptions on  $(d_m)_{m \geq 1}$  in (3.22) combined with  $\mathbb{E}[Z(\xi)] \neq 0$  ensure that for all  $\xi$  in  $\Xi$ ,  $M(\xi)$  is almost surely finite. Then, for all  $\xi$  in  $\Xi$ , the estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  is well defined. Applying Proposition 3.2 for each  $Z(\xi)$  with  $\delta = \lambda/\#\Xi$  and  $\epsilon = \frac{\tau}{2+\tau}$ , we obtain (3.23).

Now let  $A(\xi) = \left\{\left|\mathbb{E}[Z(\xi)] - \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]\right| \leq \epsilon |\mathbb{E}[Z(\xi)]|\right\}$ . By (3.23),  $\mathbb{P}(A(\xi)) \geq 1 - \frac{\lambda}{\#\Xi}$  and by a union bound argument,  $\mathbb{P}(\cap_{\xi \in \Xi} A(\xi)) \geq 1 - \lambda$ . To prove that  $\hat{\xi}$  satisfies (3.21), it remains to prove that  $\cap_{\xi \in \Xi} A(\xi)$  implies  $\mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \tau |\mathbb{E}[Z(\xi^*)]|$ . In what follows we suppose that  $\cap_{\xi \in \Xi} A(\xi)$  holds. Since  $\epsilon < 1$ ,  $\mathbb{E}[Z(\xi)]$ ,  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  and  $\overline{Z(\xi)}_{m(\xi)}$  have the same sign, that we denote by  $s(\xi)$ . Since  $A(\xi^*) \cap A(\hat{\xi})$  holds, we have

$$\begin{aligned} \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] &\leq \mathbb{E}[Z(\xi^*)] - \frac{\hat{\mathbb{E}}_{m(\hat{\xi})}[Z(\hat{\xi})]}{1 + s(\hat{\xi})\epsilon} \leq \mathbb{E}[Z(\xi^*)] - \frac{\hat{\mathbb{E}}_{m(\xi^*)}[Z(\xi^*)]}{1 + s(\hat{\xi})\epsilon} \\ &\leq \mathbb{E}[Z(\xi^*)] - \frac{1 - s(\xi^*)\epsilon}{1 + s(\hat{\xi})\epsilon} \mathbb{E}[Z(\xi^*)] \\ &= \frac{\epsilon(s(\xi^*) + s(\hat{\xi}))}{1 + s(\hat{\xi})\epsilon} \mathbb{E}[Z(\xi^*)]. \end{aligned}$$

Then we deduce

$$(3.24) \quad \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \frac{2\epsilon}{1 - \epsilon} |\mathbb{E}[Z(\xi^*)]| = \tau |\mathbb{E}[Z(\xi^*)]|,$$

which ends the proof.  $\square$

**Remark 3.7.** *If  $\mathbb{E}[Z(\xi^*)] > 0$ , we can prove that  $s(\hat{\xi}) = s(\xi^*) = 1$ , so that the inequality (3.24) becomes*

$$\mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \frac{2\epsilon}{1 + \epsilon} |\mathbb{E}[Z(\xi^*)]|.$$

*Therefore, we can set  $\epsilon = \frac{\tau}{2-\tau}$  in Algorithm 3.1 to lower the complexity and still guarantee that  $\hat{\xi}$  satisfies (3.21).*

Algorithm 3.1 provides for each random variable an estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  that satisfies (3.23). However, as will be illustrated later, this algorithm tends to use many samples for variables with a low expectation in absolute value. We propose in the next subsection an adaptive algorithm avoiding this drawback by using confidence intervals, which results in a lower overall complexity.

### 3.3.2 Adaptive Algorithm

The idea of the adaptive algorithm is to successively increase the number of samples  $m(\xi)$  of a subset of random variables  $Z(\xi)$  that are selected based on confidence intervals of  $\mathbb{E}[Z(\xi)]$  deduced from the concentration inequality of Theorem 3.1. This algorithm follows the main lines of the racing algorithms [79, section 4]. However racing algorithms do not allow to sample again a random variable discarded in an earlier step of the algorithm. The adaptive algorithm presented hereafter allows it.

In order to present this adaptive algorithm, for each  $\xi$ , we introduce the confidence interval  $[\beta_{m(\xi)}^-(\xi), \beta_{m(\xi)}^+(\xi)]$ , with

$$(3.25) \quad \beta_{\xi, m(\xi)}^- = \overline{Z(\xi)}_{m(\xi)} - c_{\xi, m(\xi)} \quad \text{and} \quad \beta_{\xi, m(\xi)}^+ = \overline{Z(\xi)}_{m(\xi)} + c_{\xi, m(\xi)}.$$

From (3.7), we have that

$$(3.26) \quad \mathbb{P} \left( \beta_{\xi, m(\xi)}^- \leq \mathbb{E}(Z(\xi)) \leq \beta_{\xi, m(\xi)}^+ \right) \geq 1 - d_{m(\xi)}.$$

We define  $\epsilon_{\xi, m(\xi)}$  by

$$\epsilon_{\xi, m(\xi)} = \frac{c_{\xi, m(\xi)}}{|\overline{Z(\xi)}_{m(\xi)}|}$$

if  $\overline{Z(\xi)}_{m(\xi)} \neq 0$ , or  $\epsilon_{\xi, m(\xi)} = +\infty$  otherwise. Letting  $s(\xi) := \text{sign}(\overline{Z(\xi)}_{m(\xi)})$ , we use as an estimate for  $\mathbb{E}[Z(\xi)]$

$$(3.27) \quad \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)] = \begin{cases} \overline{Z(\xi)}_{m(\xi)} - \epsilon_{\xi, m(\xi)} s(\xi) c_{\xi, m(\xi)} & \text{if } \epsilon_{\xi, m(\xi)} < 1, \\ \overline{Z(\xi)}_{m(\xi)} & \text{otherwise.} \end{cases}$$

If  $\epsilon_{\xi, m(\xi)} < 1$ , we note that

$$\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)] = (\overline{Z(\xi)}_{m(\xi)} \mp c_{\xi, m(\xi)}) (1 \pm s(\xi) \epsilon_{\xi, m(\xi)}),$$

so that

$$(3.28) \quad \frac{\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]}{1 \pm s(\xi) \epsilon_{\xi, m(\xi)}} = \beta_{\xi, m(\xi)}^\mp.$$

The adaptive algorithm is described in Algorithm 3.2. At each iteration  $n$ , one sample of  $Z(\xi)$  is drawn for each  $\xi$  in a subset  $\Xi_n$  selected according to (3.29). In the next proposition, we prove that the algorithm returns a solution to (3.21) under suitable assumptions.

**Proposition 3.8.** *Let  $(\tau, \lambda) \in (0, 1)^2$ . We assume that  $(d_m)_{m \geq 1}$  is a positive sequence that satisfies*

$$(3.30) \quad \sum_{m=1}^{+\infty} d_m \leq \frac{\lambda}{\#\Xi} \quad \text{and} \quad \log(3/d_m)/m \xrightarrow{m \rightarrow +\infty} 0.$$

Moreover, we assume that, for all  $\xi$  in  $\Xi$ ,  $Z(\xi)$  is a bounded random variable with  $\mathbb{E}[Z(\xi)] \neq 0$ . Then, it holds almost surely that Algorithm 3.2 stops and  $\hat{\xi}$  satisfies (3.21).

---

**Algorithm 3.2** Adaptive PAC Algorithm in Relative Precision
 

---

**Require:**  $\tau, \lambda, \{Z(\xi)\}_{\xi \in \Xi}$ .

**Ensure:**  $\hat{\xi}$ .

- 1: Set  $n = 0, \Xi_0 = \Xi, \epsilon_{\xi,0} = +\infty$  and  $m(\xi) = 0$  for all  $\xi \in \Xi$ .
- 2: **while**  $\#\Xi_n > 1$  **and**  $\max_{\xi \in \Xi_n} \epsilon_{\xi,m(\xi)} > \frac{\tau}{2+\tau}$  **do**
- 3:   **for all**  $\xi \in \Xi_n$  **do**
- 4:     Sample  $Z(\xi)$ , increment  $m(\xi)$  and update  $\epsilon_{\xi,m(\xi)}$ .
- 5:     Build the estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  using (3.27).
- 6:   **end for**
- 7:   Increment  $n$  and put in  $\Xi_n$  every  $\xi \in \Xi$  such that

$$(3.29) \quad \beta_{\xi,m(\xi)}^+ \geq \max_{\nu \in \Xi} \beta_{\nu,m(\nu)}^-(\nu).$$

8: **end while**

9: Select  $\hat{\xi}$  such that

$$\hat{\xi} \in \arg \max_{\xi \in \Xi_n} \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)].$$


---

*Proof.* Let  $m_n(\xi)$  denote the number of samples of  $Z(\xi)$  at iteration  $n$  of the algorithm. We first prove by contradiction that Algorithm 3.2 stops almost surely. Let us suppose that Algorithm 3.2 does not stop with probability  $\eta > 0$ , that means

$$(3.31) \quad \mathbb{P}\left(\forall n > 0, \#\Xi_n > 1 \quad \text{and} \quad \max_{\xi \in \Xi_n} \epsilon_{\xi,m_n(\xi)} > \frac{\tau}{2+\tau}\right) = \eta > 0.$$

Since  $(\Xi_n)_{n \geq 1}$  is a sequence from a finite set, we can extract a constant subsequence, still denoted  $(\Xi_n)_{n \geq 1}$ , equal to  $\Xi^* \subset \Xi$ , with  $\Xi^* \neq \emptyset$  such that

$$(3.32) \quad \mathbb{P}\left(\forall n > 1, \max_{\xi \in \Xi^*} \epsilon_{\xi,m_n(\xi)} > \frac{\tau}{2+\tau}\right) \geq \eta > 0.$$

Since  $\log(3/d_m)/m \rightarrow 0$  as  $m \rightarrow +\infty$  and  $\mathbb{E}[Z(\xi)] \neq 0$  for all  $\xi$ , we have that  $\epsilon_{\xi,m} \xrightarrow[m \rightarrow +\infty]{a.s.} 0$  for all  $\xi$  in  $\Xi$ . Yet, since at each iteration  $n$  (from the subsequence), we increase  $m_n(\xi)$  for all  $\xi$  in  $\Xi^*$ , we have that  $m_n(\xi) \rightarrow \infty$  as  $n \rightarrow \infty$  for all  $\xi \in \Xi^*$ . Therefore,  $\lim_{n \rightarrow +\infty} \max_{\xi \in \Xi^*} \epsilon_{\xi,m_n(\xi)} = 0$  holds almost surely, which contradicts (3.32).

We now prove that  $\hat{\xi}$  satisfies (3.21). For clarity, we remove the index  $n$  from  $m_n(\xi)$ . Defining  $A(\xi) = \left\{|\overline{Z(\xi)}_{m(\xi)} - \mathbb{E}[Z(\xi)]| \leq c_{\xi,m(\xi)}\right\}$  for all  $\xi$  in  $\Xi$ , we proceed as in (3.13) to obtain

$$\mathbb{P}(A(\xi)) \geq 1 - \lambda/\#\Xi.$$

Thus, by a union bound argument

$$\mathbb{P}(\cap_{\xi \in \Xi} A(\xi)) \geq 1 - \lambda.$$

It remains to prove that  $\cap_{\xi \in \Xi} A(\xi)$  implies  $|\mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})]| \leq \tau |\mathbb{E}[Z(\xi^*)]|$  in order to prove that  $\hat{\xi}$  satisfies (3.21). In the rest of the proof, we suppose that  $\cap_{\xi \in \Xi} A(\xi)$  holds. First, for all  $\xi \notin \Xi_n$ , using (3.25), we have

$$(3.33) \quad \mathbb{E}[Z(\xi)] \leq \beta_{\xi,m(\xi)}^+ < \max_{\nu \in \Xi} \beta_{\nu,m(\nu)}^- \leq \max_{\nu \in \Xi} \mathbb{E}[Z(\nu)] = \mathbb{E}[Z(\xi^*)],$$

that implies  $\xi^* \in \Xi_n$ . If the stopping condition is  $\#\Xi_n = 1$ , we then have  $\hat{\xi} = \xi^*$ . If the stopping condition is  $\max_{\xi \in \Xi_n} \epsilon_{\xi, m(\xi)} \leq \frac{\tau}{2+\tau} < 1$ , it means that, for all  $\xi \in \Xi_n$ ,  $\epsilon_{\xi, m(\xi)} \leq \frac{\tau}{2+\tau} < 1$ . Then for all  $\xi \in \Xi_n$ , using Proposition 3.2 with  $\epsilon = \epsilon_{\xi, m(\xi)} < 1$  and  $\delta = \lambda/\#\Xi < 1$  and the fact that  $\cap_{\xi \in \Xi} A(\xi)$  holds, we obtain that the estimate  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  satisfies

$$(3.34) \quad \left| \hat{\mathbb{E}}_{m(\xi)}[Z(\xi)] - \mathbb{E}[Z(\xi)] \right| \leq \epsilon_{\xi, m(\xi)} |\mathbb{E}[Z(\xi)]|.$$

We have that  $\epsilon_{\xi, m(\xi)} < 1$  and (3.28) hold for all  $\xi \in \Xi_n$ . In particular, since  $\hat{\xi}, \xi^* \in \Xi_n$  we get

$$\begin{aligned} \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] &\leq \mathbb{E}[Z(\xi^*)] - \frac{\hat{\mathbb{E}}_{m(\hat{\xi})}[Z(\hat{\xi})]}{1 + s(\hat{\xi})\epsilon_{\hat{\xi}, m(\hat{\xi})}} \leq \mathbb{E}[Z(\xi^*)] - \frac{\hat{\mathbb{E}}_{m(\xi^*)}[Z(\xi^*)]}{1 + s(\hat{\xi})\epsilon_{\hat{\xi}, m(\hat{\xi})}} \\ &\leq \mathbb{E}[Z(\xi^*)] - \frac{1 - s(\xi^*)\epsilon_{\xi^*, m(\xi^*)}}{1 + s(\hat{\xi})\epsilon_{\hat{\xi}, m(\hat{\xi})}} \mathbb{E}[Z(\xi^*)] \\ &= \frac{s(\xi^*)\epsilon_{\xi^*, m(\xi^*)} + s(\hat{\xi})\epsilon_{\hat{\xi}, m(\hat{\xi})}}{1 + s(\hat{\xi})\epsilon_{\hat{\xi}, m(\hat{\xi})}} \mathbb{E}[Z(\xi^*)]. \end{aligned}$$

Then we deduce

$$(3.35) \quad \mathbb{E}[Z(\xi^*)] - \mathbb{E}[Z(\hat{\xi})] \leq \frac{2\tau/(2+\tau)}{1 - \tau/(2+\tau)} |\mathbb{E}[Z(\xi^*)]| = \tau |\mathbb{E}[Z(\xi^*)]|,$$

which ends the proof.  $\square$

**Remark 3.9.** As for Algorithm 3.1 (see Remark 3.7), if  $\mathbb{E}[Z(\xi^*)] > 0$ , we can set  $\epsilon = \frac{\tau}{2-\tau}$  in Algorithm 3.2 to lower the complexity and still guarantee that  $\hat{\xi}$  satisfies (3.21).

**Remark 3.10.** A variant of Algorithm 3.2 using batch sampling would generate several samples of  $Z(\xi)$  at step 4. The result of Proposition 3.8 also holds for the algorithm with batch sampling. An optimal choice of the number of samples should depend on sampling costs.

## 3.4 Numerical Results

In this section, we propose a numerical study of the behaviour of our algorithms on a toy example. We consider the set of random variables  $Z(\xi) = f(\xi) + U(\xi)$ ,  $\xi \in \Xi$ , where  $f(\xi) = \sin(\xi) + \sin(10\xi/3)$ , the  $U(\xi)$  are i.i.d. uniform random variables over  $(-1/20, 1/20)$ , and  $\Xi = \{3 + 4i/100 : 0 \leq i \leq 100\}$ . The numerical results are obtained with the sequence  $(d_m)_{m \geq 1}$  defined by (3.16) with  $p = 2$ . We set  $\tau = 0.1$  and  $\lambda = 0.1$ .

We first compare our algorithms with two existing ones. The first one is the Median Elimination (ME) algorithm (see [37] for a description of the algorithm), that solves



problem (3.2). We take  $\tau_{abs} = \tau|\mathbb{E}[Z(\xi^*)]|$  to ensure ME algorithm provides a solution that also guarantees (3.3). Of course, this is not feasible in practice without knowing the solution of the optimization problem or at least a bound of  $|\mathbb{E}[Z(\xi^*)]|$ . The second algorithm which we compare to our algorithms is the UCB-V Algorithm (see [1, section 3.1]). It consists in only resampling the random variable whose confidence interval has the highest upper bound. To do so, we replace Steps 3 to 6 of Algorithm 3.2 by:

$$\text{Compute } \xi^+ = \arg \max_{\xi \in \Xi_n} \beta_{\xi, m(\xi)}^+,$$

Sample  $Z(\xi^+)$ , increment  $m(\xi^+)$  and update  $\epsilon_{\xi^+, m(\xi^+)}.$

We choose these algorithms to perform the comparison because i) ME Algorithm ensures theoretical guarantees similar to ours (although in absolute precision) and ii) the UCB-V Algorithm is optimal, in a sense that we will define later, for solving the optimization problem (3.1).

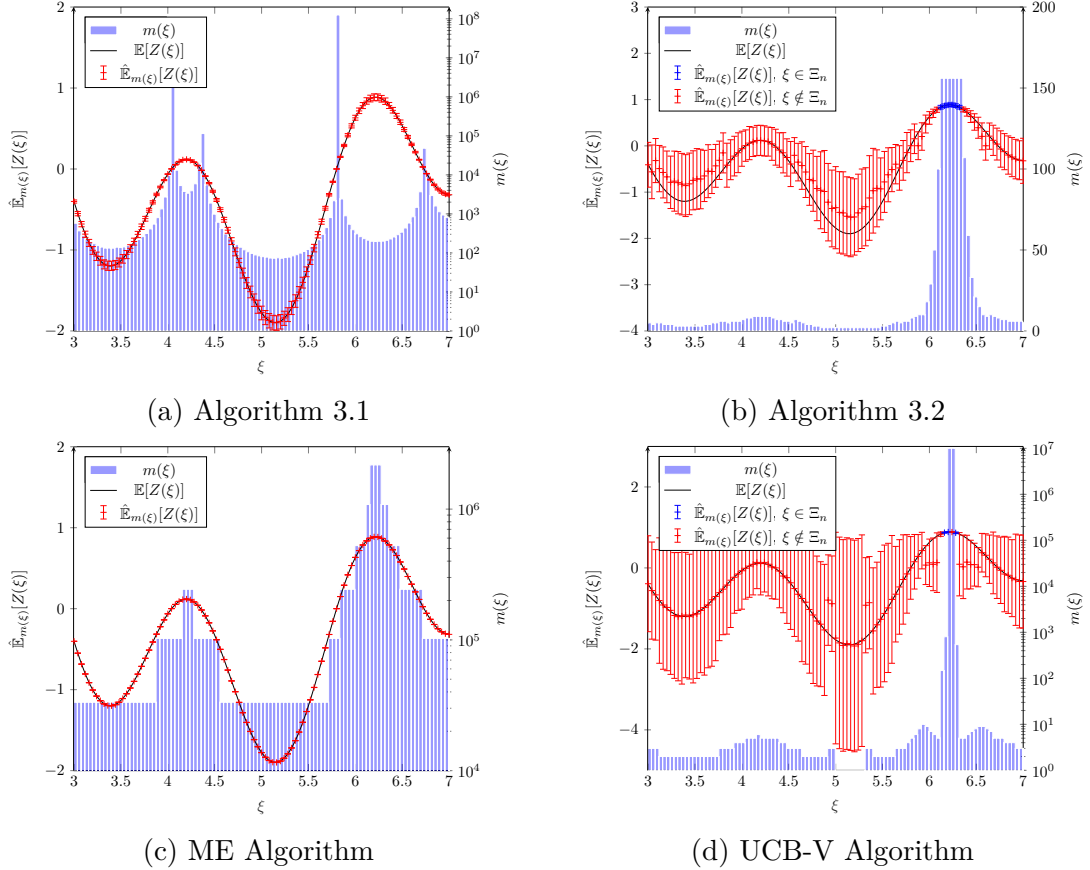


Figure 3.1: Final state of each algorithm after one run with  $\tau = 0.1$ ,  $\lambda = 0.1$  and  $\tau_{abs} = \tau|\mathbb{E}[Z(\xi^*)]|$  for ME Algorithm. Left scale : values of the estimates  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  together with the associated confidence intervals of level  $1 - d_{m(\xi)}$ . Right scale : values of  $m(\xi)$ .

We illustrate on Figure 3.1 the behavior of algorithms. The results that we show on Figure 3.1 are the ones of a single run of each algorithm. On the left scale, we plot the estimates  $\hat{\mathbb{E}}_{m(\xi)}[Z(\xi)]$  as defined in (3.27) and the associated confidence intervals

$[\beta_{\xi, m(\xi)}^-, \beta_{\xi, m(\xi)}^+]$  of level  $1 - d_{m(\xi)}$  given by (3.26). The estimates and confidence intervals for  $\xi \in \Xi_n$  are drawn in blue, while the ones for  $\xi \notin \Xi_n$  are drawn in red. On the right scale, we plot the number of samples  $m(\xi)$  generated for each  $\xi \in \Xi$ . We observe that Algorithm 3.1 samples too much the random variables with low expectation in absolute value. This is responsible for the three peaks on  $m(\xi)$  observed on Figure 3.1a. Algorithm 3.2 avoids this drawback as it does not try to reach the condition  $\epsilon_{\xi, m(\xi)} < 1$  for all random variables. The UCB-V algorithm samples mostly the two random variables with highest expectations (more than 99% of the samples are drawn from these random variables). Other random variables are not sufficiently often sampled for reaching rapidly the stopping condition based on confidence intervals. The Median Elimination Algorithm oversamples all random variables in comparison with other algorithms.

**Complexity.** To perform a quantitative comparison with existing algorithms in the case of costly sampling, a relevant complexity measure is the total number of samples generated after a single run of the algorithm

$$\mathcal{M} = \sum_{\xi \in \Xi} m(\xi).$$

Table 3.1 shows the average complexity  $\mathbb{E}(\mathcal{M})$  estimated using 30 independent runs of each algorithm. We observe that the expected complexity of Algorithm 3.2 is far below the one of the other algorithms. It means that, for the complexity measure  $\mathbb{E}(\mathcal{M})$ , the adaptive algorithm we have proposed performs the best.

ME Alg.	Alg. 3.1	Alg. 3.2	UCB-V Alg.
$2.0 \cdot 10^7$	$1.4 \cdot 10^8$	$1.9 \cdot 10^3$	$1.9 \cdot 10^8$

Table 3.1: Average complexity  $\mathbb{E}(\mathcal{M})$ , estimated using 30 runs for each algorithm, with  $\tau = 0.1$ ,  $\lambda = 0.1$  and  $\tau_{abs} = \tau |\mathbb{E}[Z(\xi^*)]|$  for ME algorithm.

We now compare the four algorithms in terms of expected runtime, that is a measure of complexity taking into account the sampling cost and the cost of all other operations performed by the algorithms. Denoting by  $t^*$  the time (assumed constant) for generating one sample from a distribution, the runtime of an algorithm is a random variable  $T = \mathcal{M}t^* + \mathcal{N}$ , where  $\mathcal{M}t^*$  is the sampling time, and  $\mathcal{N}$  is the (random) time taken by all other operations. The expected runtime is then  $\mathbb{E}(T) = \mathbb{E}(\mathcal{M})t^* + \mathbb{E}(\mathcal{N})$ . From the values of  $\mathbb{E}(\mathcal{N})$  and  $\mathbb{E}(\mathcal{M})$ , estimated over 30 runs of the algorithms, we deduce Table 3.2, which shows the average runtime  $\mathbb{E}(T)$  for different values of  $t^*$ . We observe that Algorithm 3.2 has the smallest average runtime whatever the sampling cost. The first line corresponds to  $\mathbb{E}(\mathcal{N})$  and shows that Algorithm 3.2 performs the best when sampling cost  $t^* = 0$  (or negligible). The impressive gain for large sampling costs  $t^*$  is due to the small value of the average number of samples  $\mathbb{E}(\mathcal{M})$  required by the algorithm.

$t^*$	ME Alg.	Alg. 3.1	Alg. 3.2	UCB-V Alg.
0	2.5911	$5.0 \cdot 10^1$	$3 \cdot 10^{-3}$	$1.2 \cdot 10^3$
$10^{-6}$	$2.0 \cdot 10^1$	$1.9 \cdot 10^2$	$3.8 \cdot 10^{-3}$	$1.4 \cdot 10^3$
$10^{-4}$	$2.0 \cdot 10^3$	$1.9 \cdot 10^4$	$3.8 \cdot 10^{-1}$	$1.4 \cdot 10^5$
$10^{-2}$	$2.0 \cdot 10^5$	$1.9 \cdot 10^6$	$3.8 \cdot 10^1$	$1.4 \cdot 10^7$
1	$2.0 \cdot 10^7$	$1.9 \cdot 10^8$	$3.8 \cdot 10^3$	$1.4 \cdot 10^9$

Table 3.2: Estimated runtime  $T$  (in seconds) for different values of  $t^*$ , with  $\tau = \lambda = 0.1$  and  $\tau_{abs} = \tau|\mathbb{E}[Z(\xi^*)]|$  for ME algorithm.

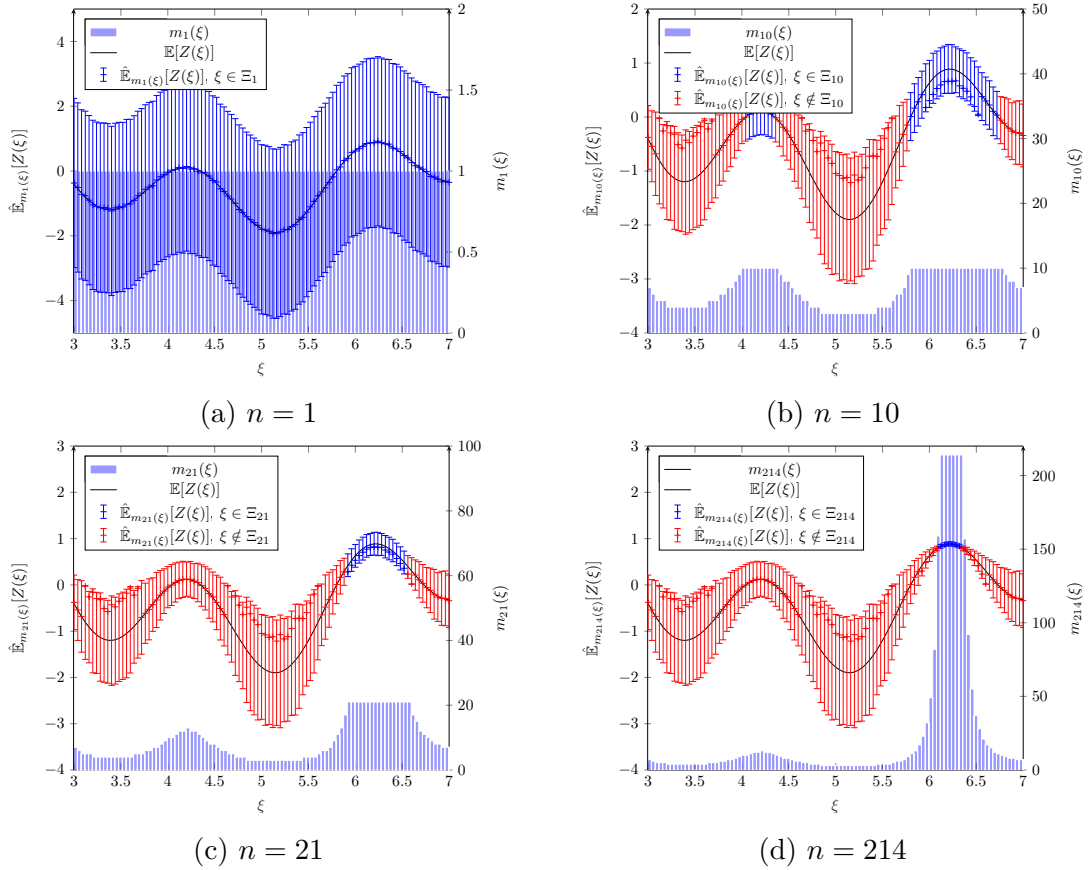


Figure 3.2: Evolution of  $\Xi_n$  and number of samples  $m_n(\xi)$  with  $n$  for Algorithm 3.2 with  $\tau = \lambda = 0.1$ .

**Behavior of Algorithm 3.2.** Now, we illustrate the behavior of Algorithm 3.2 on Figure 3.2 and show the evolution with  $n$  of  $\Xi_n$  and  $m_n(\xi)$  for a single run of Algorithm 3.2, where  $m_n(\xi)$  denotes the total number of samples from  $Z(\xi)$  generated from iteration 1 to iteration  $n$ . When  $n = 1$ , the algorithm has sampled every random variable once, which is not enough to distinguish some confidence intervals. So  $\Xi_1$  is equal to  $\Xi$ . When  $n = 10$ , some confidence intervals can be distinguished and the algorithm has identified two groups of values where a quasi-maximum could be. These two groups correspond to the two groups of random variables in  $\Xi_{10}$ . When  $n = 21$ , the algorithm has identified the main peak of the function. However, the values of  $\epsilon_{\xi, m(\xi)}$  for  $\xi$  in  $\Xi_{21}$  are not small enough for the algorithm to stop. Then the algorithm continues sampling the random variables in  $\Xi_n$ , updating  $\Xi_n$  when it

is necessary.  $\epsilon_{\xi, m(\xi)}$  for  $\xi$  in  $\Xi_n$  decreases since  $m(\xi)$  is increasing for these values of  $\xi$  and the algorithm stops at  $n = 214$  when  $\max_{\xi \in \Xi_{211}} \epsilon_{\xi, m(\xi)} < \frac{\tau}{2+\tau}$ .

Figure 3.3 shows the influence of  $\tau$  and  $\lambda$  on the average complexity  $\mathbb{E}(\mathcal{M})$  of Algorithm 3.2. We observe that  $\tau$  has a much bigger impact than  $\lambda$ . This observation is consistent with the impact of  $\epsilon = \tau/(2 + \tau)$  and  $\delta = \lambda/\#\Xi$  on the expected number of sampling  $\mathbb{E}(M)$  to build an estimate  $\hat{\mathbb{E}}_M[Z]$  of  $\mathbb{E}[Z(\xi)]$  with relative precision  $\epsilon$  with probability  $1 - \delta$  (see Remark 3.5).

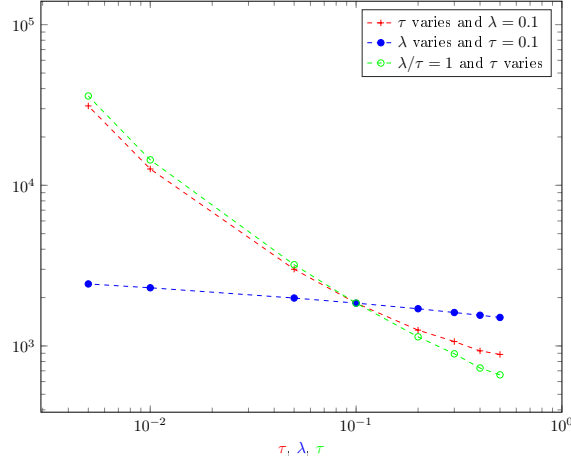


Figure 3.3: Average complexity  $\mathbb{E}(\mathcal{M})$  of Algorithm 3.2 with respect to  $\tau$  and  $\lambda$  (in log-log scale).

### 3.A Intermediate Results

Here we provide intermediate results used thereafter for the proof of Proposition 3.4 in section 3.B. We first recall a version of Bennett's inequality from [1, Lemma 5].

**Lemma 3.11.** *Let  $U$  be a random variable defined on  $(\Omega, \mathcal{F}, \mathbb{P})$  such that  $U \leq b$  almost surely, with  $b \in \mathbb{R}$ . Let  $U_1, \dots, U_m$  be i.i.d. copies of  $U$  and  $\bar{U}_\ell = \frac{1}{\ell} \sum_{i=1}^\ell U_i$ . For any  $x > 0$ , it holds, with probability at least  $1 - \exp(-x)$ , simultaneously for all  $1 \leq \ell \leq m$*

$$(3.36) \quad \ell \left( \bar{U}_\ell - \mathbb{E}[U] \right) \leq \sqrt{2m\mathbb{E}[U^2]x} + b_+x/3,$$

with  $b_+ = \max(0, b)$ .

Now, the following result provides a bound with high probability for the estimated variance of an i.i.d. sequence of bounded random variables.

**Lemma 3.12.** *Let  $X$  be a bounded random variable defined on  $(\Omega, \mathcal{F}, \mathbb{P})$ , such that  $a \leq X \leq b$  almost surely, with  $a < b$  two real numbers. Let  $X_1, \dots, X_m$  be i.i.d. copies of  $X$  and  $\bar{V}_m = \frac{1}{m} \sum_{i=1}^m (X_i - \bar{X}_m)^2$  where  $\bar{X}_m = \frac{1}{m} \sum_{i=1}^m X_i$ . Then, for any*

$x > 0$

$$(3.37) \quad \mathbb{P} \left( \bar{V}_m \leq \mathbb{V}[X] + \sqrt{2\mathbb{V}[X] \frac{(b-a)^2 x}{m}} + \frac{x(b-a)^2}{3m} \right) \geq 1 - \exp(-x).$$

*Proof.* Let us define  $U = (X - \mathbb{E}[X])^2$  which satisfies  $U \leq (b-a)^2$  almost surely. Applying Lemma 3.11 with  $U$  defined previously with  $\ell = m$  gives for any  $x > 0$

$$\mathbb{P} \left( m(\bar{U}_m - \mathbb{E}[U]) \leq \sqrt{2m\mathbb{E}[U^2]x} + \frac{x(b-a)^2}{3} \right) \geq 1 - \exp(-x).$$

Moreover, as  $\bar{U}_m = \bar{V}_m + (\bar{X}_m - \mathbb{E}[X])^2$  and using the boundedness of  $U$  we get

$$\mathbb{P} \left( \bar{V}_m \leq \mathbb{E}[U] + \sqrt{2\mathbb{E}[U] \frac{(b-a)^2 x}{m}} + \frac{x(b-a)^2}{3m} \right) \geq 1 - \exp(-x),$$

which ends the proof since  $\mathbb{E}[U] = \mathbb{V}[X]$ .  $\square$

We recall a second result in the line of [78, Lemma 3].

**Lemma 3.13.** *Let  $q, k$  be positive real numbers. If  $t > 0$  is a solution of*

$$(3.38) \quad \frac{\log qt}{t} = k,$$

*then*

$$(3.39) \quad t \leq \frac{2}{k} \log \frac{2q}{k}.$$

*Moreover, if  $t'$  is such that*

$$(3.40) \quad t' \geq \frac{2}{k} \log \frac{2q}{k},$$

*then*

$$(3.41) \quad \frac{\log qt'}{t'} \leq k.$$

*Proof.* Let  $t > 0$  be a solution of (3.38). Since the function  $\log$  is concave, it holds for all  $s > 0$

$$kt = \log(qt) \leq \log(qs) + \frac{t-s}{s}.$$

In particular, for  $s = \frac{2}{k} > 0$  we get

$$(3.42) \quad t \leq \frac{2}{k} \left( \log \frac{2q}{k} - 1 \right) \leq \frac{2}{k} \log \frac{2q}{k},$$

which yields (3.39).

Now, let  $\varphi : s \mapsto \frac{\log(qs)}{s}$  defined for  $s > 0$ . This function is continuous, strictly increasing on  $(0, \frac{e}{q}]$  and strictly decreasing on  $[\frac{e}{q}, \infty)$  so it admits a maximum at  $t = \frac{e}{q}$ . The existence of a solution  $t > 0$  of (3.38) implies  $k \leq \frac{q}{e}$ . If  $k = \frac{q}{e}$  then  $t = \frac{e}{q}$  and  $\varphi(t)$  is the maximum of  $\varphi$ . For any  $t' > 0$ , in particular satisfying (3.40), we have  $\varphi(t') \leq \varphi(t) = k$  which is (3.41). If  $0 < k < \frac{q}{e}$ , there are two solutions  $t_1, t_2$  to (3.38) such that  $0 < t_1 < \frac{e}{q} < t_2$ . By (3.39) and (3.40) we have  $t' \geq t_2 > \frac{e}{q}$  and since  $\varphi$  is strictly decreasing on  $[\frac{e}{q}, \infty)$  it holds  $\varphi(t') \leq \varphi(t_2) = k$ , that is (3.41).  $\square$

### 3.B Proof of Proposition 3.4

Let us define the two events  $A = \bigcap_{m \geq 1} A_m$  and  $B = \bigcap_{m \geq 1} B_m$  with

$$A_m = \left\{ \bar{V}_m \leq \sigma^2 + \sqrt{2\sigma^2(b-a)^2 \log(3/d_m)/m + \log(3/d_m)(b-a)^2/3m} \right\},$$

and

$$B_m = \left\{ |\bar{Z}_m - \mu| \leq c_m \right\}.$$

Applying Lemma 3.12 with  $x = \log(3/d_m)$  for  $A_m, m \geq 1$  together with a union bound argument leads to  $\mathbb{P}(A) \geq 1 - \delta/3$ . Similarly, using a union bound argument and Theorem 3.1 with  $x = \log(3/d_m)$ , for  $B_m, m \geq 1$ , gives  $\mathbb{P}(B) \geq 1 - \delta$ . By gathering these two results we have

$$(3.43) \quad \mathbb{P}(A \cap B) \geq 1 - (\mathbb{P}(A^c) + \mathbb{P}(B^c)) \geq 1 - \frac{4\delta}{3},$$

where  $A^c$  and  $B^c$  correspond respectively to the complementary events of  $A$  and  $B$ . It remains to prove that  $A \cap B$  implies

$$(3.44) \quad M \leq \left\lceil \frac{2}{\nu} \left[ \log\left(\frac{3}{\delta c}\right) + p \log\left(\frac{2p}{\nu}\right) \right] \right\rceil,$$

which will prove (3.17). In what follows, we suppose that  $A \cap B$  holds.

First we derive an upper bound for  $\bar{V}_m$ . Since  $A$  holds, we have

$$(3.45) \quad \bar{V}_m \leq \sigma^2 + \sqrt{2\sigma^2(b-a)^2 \log(3/d_m)/m + \log(3/d_m)(b-a)^2/3m}.$$

Lemma 3.13 with  $k = \frac{\sigma^2}{p(b-a)^2}$  and  $q = \left(\frac{3}{\delta c}\right)^{1/p}$  gives for any integer  $m \geq M_{\sigma^2}$

$$(3.46) \quad \frac{(b-a)^2}{m} \log \frac{3}{d_m} \leq \sigma^2,$$

where

$$M_{\sigma^2} = \frac{2(b-a)^2}{\sigma^2} \left( p \log \left( \frac{2p(b-a)^2}{\sigma^2} \right) + \log \left( \frac{3}{c\delta} \right) \right).$$

Again, Lemma 3.13 with  $k = \frac{\epsilon^2 \mu^2}{p(b-a)^2}$  and  $q = \left(\frac{3}{\delta c}\right)^{1/p}$  gives for any integer  $m \geq M_{\epsilon^2 \mu^2}$

$$(3.47) \quad \frac{(b-a)^2}{m} \log \frac{3}{d_m} \leq \epsilon^2 \mu^2,$$

where

$$M_{\epsilon^2\mu^2} = \frac{2(b-a)^2}{\epsilon^2\mu^2} \left( p \log \left( \frac{2p(b-a)^2}{\epsilon^2\mu^2} \right) + \log \left( \frac{3}{c\delta} \right) \right).$$

For all  $m \geq \min(M_{\sigma^2}, M_{\epsilon^2\mu^2})$ , i.e.  $m \geq M_{\sigma^2}$  or  $m \geq M_{\epsilon^2\mu^2}$ , we obtain from (3.45) and (3.46), or (3.45) and (3.47), that

$$(3.48) \quad \bar{V}_m \leq (1 + \sqrt{2} + 1/3) \max(\sigma^2, \epsilon^2\mu^2).$$

In what follows, we define  $\underline{M} = \min(M_{\sigma^2}, M_{\epsilon^2\mu^2})$ . Now, we deduce from (3.48) an upper bound for  $c_m$ . By definition,

$$c_m = \sqrt{\frac{2\bar{V}_m \log(3/d_m)}{m}} + \sqrt{\frac{9(b-a)^2 \log(3/d_m)^2}{m^2}},$$

then for all integer  $m \geq \underline{M}$  and using either (3.46), or (3.47), we have

$$(3.49) \quad c_m \leq \sqrt{\frac{\alpha \log(3/d_m)}{m}},$$

with  $\alpha := (\sqrt{2 + 2\sqrt{2}} + 2/3 + 3)^2 \max(\sigma^2, \epsilon^2\mu^2)$ .

Now, using (3.49), we seek a bound for  $M$ , the smallest integer such that  $c_M \leq \epsilon|\bar{Z}_M|$ . To that aim, let us introduce the integer  $M^*$ ,

$$(3.50) \quad M^* = \min \left\{ m \in \mathbb{N}^* : m \geq \underline{M}, \sqrt{\frac{\alpha \log(3/d_m)}{m}} \leq \frac{\epsilon|\mu|}{1+\epsilon} \right\},$$

and the integer valued random variable  $M^+$

$$(3.51) \quad M_+ = \min \left\{ m \in \mathbb{N}^* : c_m \leq \frac{\epsilon|\mu|}{1+\epsilon} \right\}.$$

If  $\underline{M} \geq M_+$  then  $M^* \geq M_+$ .

Otherwise,  $\underline{M} < M_+$  and we have  $M_+ = \min \left\{ m \geq \underline{M} : c_m \leq \frac{\epsilon|\mu|}{1+\epsilon} \right\}$ . Moreover, as (3.49) holds for all  $m \geq \underline{M}$ , we get the inclusion

$$\left\{ m \in \mathbb{N}^* : m \geq \underline{M}, \sqrt{\frac{\alpha \log(3/d_m)}{m}} \leq \frac{\epsilon|\mu|}{1+\epsilon} \right\} \subset \left\{ m \in \mathbb{N}^* : m \geq \underline{M}, c_m \leq \frac{\epsilon|\mu|}{1+\epsilon} \right\}.$$

Taking the min leads again to  $M^* \geq M_+$ . Moreover, since  $B$  holds,  $|\mu| - c_{M_+} \leq |\bar{Z}_{M_+}|$  and using (3.51) it implies that  $c_{M_+} \leq \epsilon|\bar{Z}_{M_+}|$ . By definition of  $M$  we get  $M_+ \geq M$ . Hence, we have  $M^* \geq M$ . To conclude the proof, it remains to find an upper bound for  $M^*$ . Applying again Lemma 3.13 with  $k = \frac{\epsilon^2\mu^2}{(1+\epsilon)^2\alpha p}$  and  $q = \left(\frac{3}{\delta c}\right)^{1/p}$  gives for any integer  $m \geq M_f$

$$(3.52) \quad \frac{\alpha \log(3/d_m)}{m} \leq \frac{\epsilon^2\mu^2}{(1+\epsilon)^2}$$

with

$$M_f = \frac{2(1+\epsilon)^2\alpha}{\epsilon^2\mu^2} \left( p \log \left( \frac{2p(1+\epsilon)^2\alpha}{\epsilon^2\mu^2} \right) + \log \left( \frac{3}{c\delta} \right) \right).$$

If  $M_f \leq \underline{M}$ , (3.50) and (3.52) imply  $M^* = \lceil \underline{M} \rceil$ , where  $\lceil \cdot \rceil$  denotes the ceil function. Otherwise  $M_f > \underline{M}$  and we obtain  $M^* \leq \lceil M_f \rceil$ . Thus, it provides the following upper bound

$$M^* \leq \max(\lceil \underline{M} \rceil, \lceil M_f \rceil) = \lceil \max(\underline{M}, M_f) \rceil.$$

Introducing  $\nu = \min \left( \frac{\max(\sigma^2, \epsilon^2\mu^2)}{(b-a)^2}, \frac{\epsilon^2\mu^2}{(1+\epsilon)^2\alpha} \right)$  we have from the definition of  $M_{\sigma^2}$ ,  $M_{\epsilon^2\mu^2}$  and  $M_f$

$$(3.53) \quad M^* \leq \left\lceil \frac{2}{\nu} \left( p \log \left( \frac{2p}{\nu} \right) + \log \left( \frac{3}{c\delta} \right) \right) \right\rceil.$$

Since  $M^* \geq M$  and  $A \cap B$  implies (3.53), we deduce that  $A \cap B$  implies (3.44), which concludes the proof of the first result.

Let us now prove the result in expectation. Let  $K := \left\lceil \frac{2}{\nu} \left( p \log \left( \frac{2p}{\nu} \right) + \log \left( \frac{3}{c\delta} \right) \right) \right\rceil$ . We first note that

$$\mathbb{E}(M) = \sum_{k=0}^{\infty} \mathbb{P}(M > k) \leq K + \sum_{k=K}^{\infty} \mathbb{P}(M > k).$$

If  $M > k$ , then  $c_k > \epsilon|\bar{Z}_k|$ . For  $k \geq K$ , we would like to prove that  $c_k > \epsilon|\bar{Z}_k|$  implies  $(A_k \cap B_k)^c$ , or equivalently that  $A_k \cap B_k$  implies  $c_k \leq \epsilon|\bar{Z}_k|$ . For  $k \geq K$ ,  $A_k$  implies (3.49) and (3.52), and therefore  $c_k \leq \frac{\epsilon|\mu|}{1+\epsilon}$ . Also,  $B_k$  implies  $|\bar{Z}_k| \geq |\mu| - c_k$ . Combining the previous inequalities, we easily conclude that  $A_k \cap B_k$  implies  $c_k \leq \epsilon|\bar{Z}_k|$ . For  $k \geq K$ , we then have  $\mathbb{P}(M > k) \leq \mathbb{P}(c_k > \epsilon|\bar{Z}_k|) \leq \mathbb{P}((A_k \cap B_k)^c) \leq \mathbb{P}(A_k^c) + \mathbb{P}(B_k^c) \leq 4d_k/3$ , and then

$$\mathbb{E}(M) \leq K + \sum_{k=K}^{\infty} 4d_k/3 \leq K + 4\delta/3,$$

which ends the proof.





# Chapter 4

## Reduced Basis Methods

This chapter surveys reduced basis methods (RBM) for parameter-dependent problems with a particular focus on parameter-dependent partial differential equations (PDEs).

### Contents

4.1	Idealized RBM . . . . .	82
4.1.1	Solution Manifold and Kolmogorov Width . . . . .	82
4.1.2	Greedy Algorithm . . . . .	83
4.1.3	Snapshot-Based Reduced Spaces . . . . .	84
4.2	Feasible Version of RBM . . . . .	85
4.2.1	Truth Approximation . . . . .	85
4.2.2	Feasible Projection . . . . .	86
4.2.3	Discrete Training Set . . . . .	86
4.2.4	Practical Greedy Algorithm . . . . .	88
4.3	RBM for Parameter-Dependent PDEs . . . . .	89
4.3.1	Weak-Formulation . . . . .	89
4.3.2	Affine Decomposition and EIM . . . . .	90
4.3.3	Error Estimation . . . . .	90

In this chapter we consider a set of parameters  $Y \subset \mathbb{R}^p$  and a bounded set  $\mathcal{O} \subset \mathbb{R}^d$  of variables (either spatial or spatio-temporal). We consider a parameter-dependent problem of the following type

$$(4.1) \quad \mathcal{S}(u(\cdot, y), y) = 0,$$

where the unknown is a real-valued function  $u : \mathcal{O} \times Y \rightarrow \mathbb{R}$ . The operator  $\mathcal{S} : \mathcal{H} \times Y \rightarrow \mathcal{W}$  is a map defined on  $\mathcal{H} \times Y$  with values in  $\mathcal{W}$ , where  $\mathcal{H} := \mathcal{H}(\mathcal{O})$  and  $\mathcal{W} := \mathcal{W}(\mathcal{O})$  are two Hilbert spaces of functions defined on  $\mathcal{O}$ , e.g.  $L^2(\mathcal{O})$  the set of square-integrable functions or the Sobolev space  $H_0^1(\mathcal{O})$ . In what follows, we lighten our notation and use  $u(y)$  to refer to  $u(\cdot, y) : \mathcal{O} \rightarrow \mathbb{R}$ .

We assume that the problem (4.1) is well-posed in  $\mathcal{H}$ , which means that for all instance  $y \in Y$ , there exists a unique solution  $u(y) \in \mathcal{H}$  that satisfies (4.1). The map  $u$  is defined on a high-dimensional set  $\mathcal{O} \times Y \subset \mathbb{R}^d \times \mathbb{R}^p$ , which causes some

difficulties when it comes to approximation. We have seen in Chapter 2 that approximation of parameter-independent high-dimensional problems is challenging. When it comes to high-dimensional parameter-dependent problems, we can imagine that other problems arise.

We focus here on a particular context to introduce these solutions. It is usually referred as the multi-query context, see e.g. [53], where many snapshots, i.e.  $u(y)$  for many different values of  $y$  in  $Y$ , are required. In such a context, one cannot only approximate the solution, e.g. using an algorithm such as Algorithm 2.3, as many times as the number of required snapshots: it would be too costly. One has to try to use already computed snapshots in order to reduce computational cost. This is what reduced basis methods (RBM) and more generally model order reduction (MOR) methods do. Our particular focus is on RBM that build a reduced space  $H_n$  using snapshots  $u(y^1), \dots, u(y^n)$  (offline phase) and then project at low computational cost any desired snapshot  $u(y)$  onto  $H_n$  (online phase). This offline-online decomposition is particularly relevant in many applications, see e.g. [93], where calculus has to be made in real-time with a possible pre-computation of some needed quantities (here the snapshots). Of course, such techniques are particularly relevant when the solution manifold  $\mathcal{M} := \{u(y), y \in Y\}$  can be well-approximated by low-dimensional subspaces of  $\mathcal{H}$ .

We discuss theoretical aspects for reduced basis methods in Section 4.1, recalling an existing ideal algorithm to build a space  $H_n$  with good approximation properties. Then in Section 4.2 we discuss practical implementation of the ideal algorithm. We finish our survey by considering in Section 4.3 the particular case of parameter-dependent PDEs.

## 4.1 Idealized RBM

We present here basic notions related to RBM from a theoretical point of view. Then we recall a first algorithm to build reduced spaces together with first approximation results.

### 4.1.1 Solution Manifold and Kolmogorov Width

As we mentioned earlier, RBM are particularly relevant for problems where the solution manifold  $\mathcal{M}$  defined below

$$(4.2) \quad \mathcal{M} = \{u(y) : y \in Y\} \subset \mathcal{H},$$

can be well-approximated by low-dimensional spaces  $H_n$ .

We assume here that  $\mathcal{M}$  is compact. Compactness can be obtained under some assumptions on  $Y$  and  $\mathcal{P}$ , see e.g. [25]. Since we will focus on linear approximation tools, we recall the notion of Kolmogorov  $n$ -width for a compact set  $\mathcal{M}$  noted

$d_n(\mathcal{M})_{\mathcal{H}}$  and defined by

$$(4.3) \quad d_n(\mathcal{M})_{\mathcal{H}} := \inf_{\dim(W)=n} \text{dist}(\mathcal{M}, W)_{\mathcal{H}},$$

where  $\text{dist}(\mathcal{M}, W)_{\mathcal{H}} := \sup_{u \in \mathcal{M}} d(u, W)_{\mathcal{H}}$  and  $d(u, W)_{\mathcal{H}} := \inf_{w \in W} \|u - w\|_{\mathcal{H}}$ . Other widths [90] can be considered when it comes to non-linear approximation, which is not our concern here, but for which we refer to [31].

Problems for which it can be interesting to use RBM are thus the ones where the Kolmogorov  $n$ -width of  $\mathcal{M}$  decreases quickly with  $n$  [90]. However, these theoretical decays can be impacted by the dimension  $d$ , which can cause troubles when  $d \gg 1$ , even for smooth functions [86]. As an example of the impact of the dimension, we recall a result from [24], that gives an estimate of the  $n$ -width of the unit ball  $\mathcal{K}$  of  $\mathcal{C}^m([-1, 1]^d)$  in  $L^\infty$ -norm

$$c_d n^{-m/d} \leq d_n(\mathcal{K})_{L^\infty} \leq C_d n^{-m/d}$$

where  $c_d$  and  $C_d$  are constants that grow exponentially with the dimension  $d$ . We thus observe that the dimension  $d$  impacts negatively the decay of the width, a phenomenon known as the curse of dimensionality, while the regularity improves this decay, phenomenon known as the blessing of regularity. However, for some other classes of more structured functions it is possible to circumvent the curse of dimensionality [24].

### 4.1.2 Greedy Algorithm

We give a first ideal algorithm for building reduced spaces. To that end, we first recall that in a Hilbert space we have

$$(4.4) \quad \inf_{w \in W} \|u - w\|_{\mathcal{H}} = \|u - P_W u\|_{\mathcal{H}}$$

where  $P_W$  stands for the  $\mathcal{H}$ -orthogonal projection onto  $W$ . We can now describe the greedy algorithm for RBM in Algorithm 4.1. We fix a finite number of steps  $n > 0$  for this algorithm. Other stopping criteria exist, such as stopping when the desired approximation precision is reached [27].

---

**Algorithm 4.1** Greedy Algorithm for Reduced Basis Methods [32]

---

- 1: Set  $H_0 = \{0\}$ .
- 2: **for**  $i = 1 \dots n$  **do**
- 3:   Select  $y^i \in Y$  such that

$$(4.5) \quad \|u(y^i) - P_{H_{i-1}} u(y^i)\|_{\mathcal{H}} \geq \gamma \sup_{y \in Y} \|u(y) - P_{H_{i-1}} u(y)\|_{\mathcal{H}}.$$

- 4:   Compute  $u(y^i)$  and update  $H_i = \text{span}\{u(y^1), \dots, u(y^i)\}$ .
  - 5: **end for**
-

Readers should notice that when  $\gamma = 1$ , Algorithm 4.1 corresponds to the strong greedy algorithm described for example in [18]. Otherwise, when  $0 < \gamma < 1$ , Algorithm 4.1 corresponds to a  $\gamma$ -weak-greedy algorithm such as the one described in [13]. We introduce the following notation

$$(4.6) \quad \sigma_n(\mathcal{M})_{\mathcal{H}} = \text{dist}(\mathcal{M}, H_n)_{\mathcal{H}},$$

where  $H_n$  has been constructed with Algorithm 4.1. We distinguish different results for the study of  $\sigma_n(\mathcal{M})_{\mathcal{H}}$ . First of all, indirect comparison results assume a particular kind of upper bound for  $d_n(\mathcal{M})_{\mathcal{H}}$  in order to derive an upper bound for  $\sigma_n(\mathcal{M})_{\mathcal{H}}$ . For a  $\gamma$ -weak-greedy algorithm when  $\mathcal{H}$  is a Hilbert space, we have the following existing result.

**Theorem 4.1.** *[32, Corollary 3.3] Assume that  $\mathcal{M}$  is a compact set included in the unit ball of  $\mathcal{H}$ . Assume moreover that, for all  $n \geq 1$ ,*

$$d_n(\mathcal{M})_{\mathcal{H}} \leq C \exp(-cn^\alpha)$$

*with  $\alpha > 0$ . Then, for all  $n \geq 1$ ,*

$$\sigma_n(\mathcal{M})_{\mathcal{H}} \leq C_1 \exp(-c_1 n^\alpha),$$

$$C_1 = \sqrt{2C}\gamma^{-1} \text{ and } c_1 = 2^{-1-2\alpha}c.$$

Second of all, direct comparisons compare directly  $d_n(\mathcal{M})_{\mathcal{H}}$  and  $\sigma_n(\mathcal{M})_{\mathcal{H}}$  for any compact manifold  $\mathcal{M}$ . Those direct comparison results are usually more pessimistic than indirect comparisons results, due to less strong assumptions. Still in the case of a  $\gamma$ -weak-greedy algorithm when  $\mathcal{H}$  is a Hilbert space, we have the following result.

**Theorem 4.2.** *[32, Corollary 3.3] For any compact set  $\mathcal{M}$  included in the unit ball of  $\mathcal{H}$  and for any  $n \geq 1$ , we have*

$$(4.7) \quad \sigma_n(\mathcal{M})_{\mathcal{H}} \leq \sqrt{2}\gamma^{-1} \min_{1 \leq m \leq n} (d_m(\mathcal{M})_{\mathcal{H}})^{\frac{n-m}{n}}.$$

*It gives in particular, for all  $n \geq 1$*

$$(4.8) \quad \sigma_{2n}(\mathcal{M})_{\mathcal{H}} \leq \sqrt{2}\gamma^{-1} \sqrt{d_n(\mathcal{M})_{\mathcal{H}}}.$$

### 4.1.3 Snapshot-Based Reduced Spaces

Algorithm 4.1 builds a reduced basis using snapshots  $u(y)$  in  $\mathcal{M}$ , that are usually ortho-normalized for practical reasons. However, nothing indicates that optimal subspace for the Kolmogorov  $n$ -width can be generated by snapshots. We recall a result here from [13] showing a relatively small loss when building of the reduced spaces using snapshots.

**Theorem 4.3.** [13, Theorem 4.1] *Let*

$$\bar{d}_n(\mathcal{M})_{\mathcal{H}} = \inf_{W \in W_n(\mathcal{M})} \sup_{y \in Y} \|u(y) - P_W u(y)\|_{\mathcal{H}},$$

where  $W_n(\mathcal{M})$  is composed of all  $n$ -dimensional spaces spanned by  $n$  snapshots from  $\mathcal{M}$ . For any compact set  $\mathcal{M}$  and any  $n \geq 0$

$$\bar{d}_n(\mathcal{M})_{\mathcal{H}} \leq (n+1)d_n(\mathcal{M})_{\mathcal{H}}.$$

Moreover, for any  $n > 1$  and  $\epsilon > 0$ , there exists a set  $\mathcal{M}_1$  such that

$$\bar{d}_n(\mathcal{M}_1)_{\mathcal{H}} \geq (n-1-\epsilon)d_n(\mathcal{M}_1)_{\mathcal{H}}.$$

When the decay of  $d_n(\mathcal{M})_{\mathcal{H}}$  is too slow, e.g. polynomial in  $n$  with a degree lower or equal to 1, building a good reduced space  $H_n$  with a greedy algorithm is not necessarily feasible. However, due to Theorem 4.2, the loss in the decay of the Kolmogorov  $n$ -width resulting from the use of snapshots to build the reduced spaces is much smaller than the loss resulting from the use of greedy algorithm. Moreover since we have  $\bar{d}_n(\mathcal{M})_{\mathcal{H}} \leq \sigma_n(\mathcal{M})_{\mathcal{H}}$ , any upper bound for  $\sigma_n(\mathcal{M})_{\mathcal{H}}$  (see e.g. Theorem 4.1) holds also for  $\bar{d}_n(\mathcal{M})_{\mathcal{H}}$ .

## 4.2 Feasible Version of RBM

Algorithm 4.1 is impossible to implement in practice for several reasons. First, the exact solution  $u$  is in general inaccessible since it resides in an infinite-dimensional Hilbert space  $\mathcal{H}$ . From this arise some limitations: 1) the impossibility to compute exactly a snapshot 2) the impossibility to provide an error estimate for  $\|u(y) - P_{H_{i-1}} u(y)\|_{\mathcal{H}}$  that takes as a reference the exact solution  $u$  and thus the impossibility to solve the optimization problem  $\max_{y \in Y} \|u(y) - P_{H_{i-1}} u(y)\|_{\mathcal{H}}$ . Other problems arise. Among others 3) the impossibility to compute an exact orthogonal projection  $P_{H_{i-1}}$  (which is an even more important limitation when  $u$  is not accessible) and 4) the difficulty to solve the optimization problem over a continuous set  $Y$ . The last point is due to the complete absence of knowledge on properties (convexity, continuity ...) of  $y \rightarrow \|u(y) - P_{H_{i-1}} u(y)\|_{\mathcal{H}}$ . To circumvent these problems, we recall a realistic framework for RBM in Section 4.2.1 before showing how it is possible to derive a weaker version of Algorithm 4.1 in Section 4.2.4.

### 4.2.1 Truth Approximation

Since the exact solution  $u$  is inaccessible, we rather use an accurate approximation  $u^{\mathcal{N}}$  of  $u$ , called the truth approximation. This approximation lives in some high-dimensional space  $\mathcal{H}^{\mathcal{N}}$ , called the truth space, and can be computed by traditional approximation methods for the approximation of the solution of high-dimensional PDEs (see Chapter 2). We assume that it approximates  $u$  at some precision  $\epsilon \in (0, 1)$ , which means that for any  $y$  in  $Y$  we have

$$(4.9) \quad \|u(y) - u^{\mathcal{N}}(y)\|_{\mathcal{H}} \leq \epsilon.$$

Two different possibilities exist for the space  $\mathcal{H}^{\mathcal{N}}$ . Either the space  $\mathcal{H}^{\mathcal{N}} \subset \mathcal{H}$  of dimension  $\mathcal{N}$  is fixed and large, i.e.  $\mathcal{N} \gg 1$ . Either the space  $\mathcal{H}^{\mathcal{N}}(y) \subset \mathcal{H}$  depends on  $y$ , e.g. through a dependence of its dimension  $\mathcal{N} = \mathcal{N}(y)$ . In the first case  $\mathcal{H}^{\mathcal{N}}$  ensures that, for any  $y$  in  $Y$ , the solution  $u(y)$  in  $\mathcal{H}$  can be well represented by elements in  $\mathcal{H}^{\mathcal{N}}$ . An example of such space is a finite element approximation space with  $\mathcal{N}$  degrees of freedom. In the second case, the dependence on  $y$  allows important computational savings when it comes to the computation of the truth approximation of some snapshot up to some given precision  $\epsilon$ , see (4.9). While more flexible, the second case is more difficult to describe in a few lines. Thus, for the sake of clarity, we stick to the first case in our presentation.

In the rest of our presentation, we assume that the exact and approximated solution coincide, which means that we take  $\epsilon \ll 1$  of the order of magnitude of the machine precision for example. It allows us to neglect the differences between the high-dimensional and high-fidelity truth approximation  $u^{\mathcal{N}}$  and the exact solution  $u$ . Considering the truth approximation rather than the exact solution is not problematic when  $\epsilon \ll 1$ . In this case we can prove in a similar way as we will prove Theorem 4.4 that the behavior of both Kolmogorov  $n$ -width if  $\mathcal{M}$  and  $\mathcal{M}^{\mathcal{N}} := \{u^{\mathcal{N}}(y) : y \in Y\}$  are similar up to a small threshold  $2\epsilon$ .

### 4.2.2 Feasible Projection

The truth approximation answers problem 1). In order to answer 3), we avoid using the exact orthogonal projection  $P_{H_{i-1}}$  for the benefit of an approximated projection  $u_{i-1} := u_{H_{i-1}}$  of  $u$ , that is easy to compute and that presents interesting properties, such as a quasi-optimality property. Quasi-optimality means that there exists a constant  $C > 0$ , such that for any  $y$  in  $Y$  we have

$$(4.10) \quad \|u(y) - u_{i-1}(y)\|_{\mathcal{H}} \leq C \|u(y) - P_{H_{i-1}}u(y)\|_{\mathcal{H}}.$$

The independence of  $C$  on  $y$  or  $n$  and the existence of such projection are important points that we will discuss in Section 4.2.4 for parameter-dependent PDEs.

### 4.2.3 Discrete Training Set

In order to answer 4), we have to come up with an optimization strategy to solve  $\max_{y \in Y} \|u(y) - P_{H_{i-1}}u(y)\|_{\mathcal{H}}$ . A commonly used technique is to discretize the solution manifold, which means considering a finite discrete set  $\Xi \subset Y$  and solve the optimization problem over this so-called training set. Thus the optimization problem becomes a discrete optimization problem which is possible to solve if we know how to compute  $\|u(y) - P_{H_{i-1}}u(y)\|_{\mathcal{H}}$  for a finite number of values of  $y$  in  $Y$  which corresponds to  $\Xi$ . Then, denoting by  $\mathcal{M}_{\Xi}$  the following discrete solution manifold

$$(4.11) \quad \mathcal{M}_{\Xi} = \{u(\xi) : \xi \in \Xi\},$$

we can derive the following result.

**Theorem 4.4.** *Assume that the map  $u : Y \rightarrow \mathcal{H}$  is Hölder-continuous with reference to  $y$ , which means*

$$\|u(y^1) - u(y^2)\|_{\mathcal{H}} \leq L\|y^1 - y^2\|_{\mathbb{R}^D}^{\alpha},$$

*for some  $\alpha \in ]0, 1]$ , some norm  $\|\cdot\|_{\mathbb{R}^D}$  on  $\mathbb{R}^D$  and a constant  $L$  independent of  $y^1$  and  $y^2$ . Moreover, assume that  $\Xi$  is an  $\epsilon$ -net of  $Y$ , which means  $\Xi$  is a finite subset of  $Y$  such that for all  $y$  in  $Y$  there exists a  $\xi$  in  $\Xi$  satisfying*

$$\|y - \xi\|_{\mathbb{R}^D} \leq \epsilon.$$

*Then we have*

$$d_n(\mathcal{M}_{\Xi})_{\mathcal{H}} \leq d_n(\mathcal{M})_{\mathcal{H}} \leq d_n(\mathcal{M}_{\Xi})_{\mathcal{H}} + 2L\epsilon^{\alpha}.$$

*Proof.* We have

$$d_n(\mathcal{M}_{\Xi})_{\mathcal{H}} = \inf_{\dim(W)=n} \sup_{\xi \in \Xi} \|u(\xi) - P_W u(\xi)\|_{\mathcal{H}} \leq d_n(\mathcal{M})_{\mathcal{H}}.$$

Then we notice that for all  $y$  in  $Y$  and all space  $W \subset \mathcal{H}$

$$\begin{aligned} \|u(y) - P_W u(y)\|_{\mathcal{H}} &\leq \|u(y) - u(\xi)\|_{\mathcal{H}} + \|u(\xi) - P_W u(\xi)\|_{\mathcal{H}} \\ &\quad + \|P_W u(\xi) - P_W u(y)\|_{\mathcal{H}}, \end{aligned}$$

for any  $\xi$  in  $\Xi$ . In particular, for  $\xi_y$  such that  $\|\xi_y - y\|_{\mathbb{R}^D} \leq \epsilon$  (that exists by assumption), it brings

$$\|u(y) - P_W u(y)\|_{\mathcal{H}} \leq 2L\epsilon^{\alpha} + \|u(\xi_y) - P_W u(\xi_y)\|_{\mathcal{H}}.$$

Taking the supremum over  $Y$  and the infimum over  $W$ , we obtain the second inequality.  $\square$

Theorem 4.4 ensures that if an  $\epsilon$ -net  $\Xi$  is dense enough in  $Y \subset \mathbb{R}^p$  we can expect similar behavior between  $d_n(\mathcal{M})_{\mathcal{H}}$  and  $d_n(\mathcal{M}_{\Xi})_{\mathcal{H}}$ , up to a threshold depending on the parameter  $\epsilon$ . However when the parametric dimension  $p$  is large, i.e.  $p \gg 1$ , the number of points of  $\epsilon$ -net of  $Y$  is too large since it grows exponentially with the dimension  $p$ : this is another phenomenon illustrating the curse of dimensionality.

To avoid using an  $\epsilon$ -net, the authors of [27] have proposed to use random training sets, introducing at the same time the notion of probabilistic greedy algorithm. This contribution will be discussed further in Chapter 5. Another contribution [60] has first investigated ways to avoid computing at each step of a greedy algorithm the error  $\|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}}$  over the whole training set  $\Xi$ : this technique reduces the computational cost of using large training sets but not necessarily the storage problem. The authors proposed in the same paper a heuristic method to avoid considering too big training sets by safety checking the approximation error of the reduced space they had built on random points outside of the training set. They came up with many numerical experiments illustrating the interest of their method. They also allowed their algorithm to remove and add points to the training set, which thus changes with the number of steps.



### 4.2.4 Practical Greedy Algorithm

Using the tools introduced in Section 4.2.1, we can now propose a realistic version of Algorithm 4.1 in the sense that it is possible to implement. This is a weak-greedy algorithm for (4.1) over  $Y$  which means that, at each step  $i$  of the algorithm, we add a snapshot  $u(y^i)$  such that  $y^i$  is the training set and satisfies

$$(4.12) \quad \|u(y^i) - P_{H_{i-1}}u(y^i)\|_{\mathcal{H}} \geq \gamma \sup_{y \in Y} \|u(y) - P_{H_{i-1}}u(y)\|_{\mathcal{H}}$$

for some fixed  $\gamma \in (0, 1)$ . An algorithm for which the parameter  $\gamma$  does not change from one step to another is called a  $\gamma$ -weak-greedy algorithm over  $Y$ .

In order to make Algorithm 4.2 entirely practical, and thus to tackle problem 3), we assume that we have access to some error estimate  $\Delta(u_{i-1}(y))$  for  $\|u(y) - u_{i-1}(y)\|_{\mathcal{H}}$  for all  $y$  in  $Y$ , which is necessary since the  $\mathcal{H}$ -norm might be hard to compute exactly. We moreover assume that this estimate satisfies

$$(4.13) \quad c_1 \|u(y) - u_{i-1}(y)\|_{\mathcal{H}} \leq \Delta(u_{i-1}(y)) \leq C_1 \|u(y) - u_{i-1}(y)\|_{\mathcal{H}},$$

for some constants  $c_1$  and  $C_1$  independent from  $y$  and  $i$ .

---

**Algorithm 4.2** Practical Greedy Algorithm for RBM

---

- 1: Set  $X_0 = \{0\}$ .
- 2: **for**  $i = 1 \dots n$  **do**
- 3:   Select  $\xi^i \in \Xi$  such that

$$(4.14) \quad \xi^i \in \arg \max_{\xi \in \Xi} \Delta(u_{i-1}(\xi)).$$

- 4:   Compute  $u^{\mathcal{N}}(\xi^i)$  and update  $H_i = \text{span}\{u^{\mathcal{N}}(\xi^1), \dots, u^{\mathcal{N}}(\xi^i)\}$ .
  - 5: **end for**
- 

The following theorem shows that Algorithm 4.2 is a weak-greedy algorithm under suitable assumptions.

**Theorem 4.5.** *Assume that  $\Xi$  is a training set and  $u = u^{\mathcal{N}}$  is the finite-dimensional solution to (4.1). Moreover assume that (4.13) and (4.10) hold. Then we have for all  $i = 1, \dots, n$  that*

$$(4.15) \quad \|u(\xi^i) - P_{H_{i-1}}u(\xi^i)\|_{\mathcal{H}} \geq \gamma \max_{\xi \in \Xi} \|u(\xi) - P_{H_{i-1}}u(\xi)\|_{\mathcal{H}},$$

where  $\gamma = \frac{c_1}{C_1 C}$ . Thus Algorithm 4.2 is a  $\gamma$ -weak-greedy algorithm for (4.1) over  $\Xi$ .

*Proof.* We take  $1 \leq i \leq n$ . We have using (4.14) and (4.13)

$$\begin{aligned} \Delta(u_{i-1}(\xi^i)) &= \max_{\xi \in \Xi} \Delta(u_{i-1}(\xi)) \\ &\geq c_1 \max_{\xi \in \Xi} \|u(\xi) - u_{i-1}(\xi)\|_{\mathcal{H}} \\ &\geq c_1 \max_{\xi \in \Xi} \|u(\xi) - P_{H_{i-1}}u(\xi)\|_{\mathcal{H}}. \end{aligned}$$

We have also from (4.10) and (4.13)

$$\begin{aligned}\Delta(u_{i-1}(\xi^i)) &\leq C_1 \|u(\xi^i) - u_{i-1}(\xi^i)\|_{\mathcal{H}} \\ &\leq C_1 C \|u(\xi^i) - P_{H_{i-1}} u(\xi^i)\|_{\mathcal{H}}\end{aligned}$$

which concludes the proof.  $\square$

Using this result and if we further satisfy

$$(4.16) \quad \max_{\xi \in \Xi} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}} \geq \gamma_1 \max_{y \in Y} \|u(y) - P_{H_{i-1}} u(y)\|_{\mathcal{H}},$$

we obtain a  $\gamma_1 \gamma$ -weak-greedy algorithm over  $Y$ . In this case, all results seen in Section 4.1.2 apply to study the consequences of approximating  $\mathcal{M}$  using the reduced space  $H_n$  built by Algorithm 4.2.

## 4.3 RBM for Parameter-Dependent PDEs

We consider the particular case of parameter-dependent PDEs. In order to directly consider practical methods, we consider the discrete solution manifold defined in (4.11) and work with  $\Xi$  rather than  $Y$ . Such parameter-dependent problems are particularly interesting since, for several types of parameter-dependent PDEs, it is possible to circumvent the curse of dimensionality mentioned before [23, 25].

### 4.3.1 Weak-Formulation

We consider the following parameter-dependent weak-formulation of a PDE: For any  $\xi$  in  $\Xi$ , the goal is to find  $u(\xi)$  in  $\mathcal{H}$  such that

$$(4.17) \quad a(u(\xi), v; \xi) = f(v; \xi), \quad \forall v \in \mathcal{H},$$

with  $\mathcal{H}$  some function space. For all  $\xi$  in  $\Xi$ ,  $a(\cdot, \cdot; \xi)$  is a bilinear form while  $f(\cdot; \xi)$  is a linear form. We assume the existence of positive constants  $\alpha(\xi) \leq \alpha^{\max} < \infty$  such that

$$(4.18) \quad a(w, v; \xi) \leq \alpha(\xi) \|w\|_{\mathcal{H}} \|v\|_{\mathcal{H}}, \quad \forall (w, v, \xi) \in \mathcal{H} \times \mathcal{H} \times \Xi.$$

We also assume the existence of constants  $\beta(\xi) \geq \beta^{\min} > 0$  such that, for all  $\xi$  in  $\Xi$

$$(4.19) \quad a(v, v; \xi) \geq \beta(\xi) \|v\|_{\mathcal{H}}^2 \geq \beta^{\min} \|v\|_{\mathcal{H}}^2.$$

In this context, the traditional truth space consists in Galerkin projections of  $u(\xi)$  onto some high-dimensional space  $\mathcal{H}^{\mathcal{N}}$ . The Galerkin projection of  $u(\xi)$  onto a space  $\mathcal{H}^{\mathcal{N}}$ , denoted by  $u^{\mathcal{N}}(\xi)$  is defined for each  $\xi$  in  $\Xi$  as the unique solution of

$$(4.20) \quad a(u^{\mathcal{N}}(\xi), v; \xi) = f(v; \xi), \quad \forall v \in \mathcal{H}^{\mathcal{N}}.$$

Galerkin projection has the following property (C ea's lemma)

$$(4.21) \quad \|u(\xi) - u^{\mathcal{N}}(\xi)\|_{\mathcal{H}} \leq \frac{\alpha^{\max}}{\beta^{\min}} \|u(\xi) - P_{\mathcal{H}^{\mathcal{N}}} u(\xi)\|_{\mathcal{H}}.$$

Having a basis for  $\mathcal{H}^{\mathcal{N}}$ , we can solve (4.20) for a fixed value of  $\xi$  by solving a high-dimensional linear system (see e.g. discrete reduced basis in [53]). We show now how to use the Galerkin projection to propose an efficient offline-online decomposition.

### 4.3.2 Affine Decomposition and Empirical Interpolation Method

We assume now that the forms  $a$  and  $f$  admit an affine decomposition of the following type

$$(4.22) \quad a(w, v; \xi) = \sum_{i=1}^{Q_a} \Theta_i^a(\xi) a_i(w, v),$$

$$(4.23) \quad f(v; \xi) = \sum_{i=1}^{Q_f} \Theta_i^f(\xi) f_i(v),$$

where  $\Theta_i^a, \Theta_i^f$  are parameter-dependent functions,  $a_i, f_i$  are parameter-independent forms and  $Q_a, Q_f$  are strictly positive integers.

Once the snapshots  $(u^\mathcal{N}(\xi^j))_{j=1}^n$  are computed, it is possible to compute the Galerkin projection  $u_n^\mathcal{N}(\xi)$  of  $u(\xi)$  onto  $H_n = \text{span} \{u^\mathcal{N}(\xi^1), \dots, u^\mathcal{N}(\xi^n)\}$  at low cost by pre-computing the matrices  $\mathbf{A}^j = [a_j(u^\mathcal{N}(\xi^i), u^\mathcal{N}(\xi^k))]_{j,k}$  for all  $j$ , the vectors  $\mathbf{f}^j = [f_j(u^\mathcal{N}(\xi^k))]_{k=1}^n$ . Using (4.22) and (4.23), we end up with a linear system

$$\begin{aligned} \mathbf{A}_n(\xi) \mathbf{u}_n(\xi) &= \mathbf{f}_n(\xi), \quad \text{where} \\ \mathbf{A}_n(\xi) &= \sum_{i=1}^{Q_a} \Theta_i^a(\xi) \mathbf{A}^i \quad \text{and} \quad \mathbf{f}_n(\xi) = \sum_{i=1}^{Q_f} \Theta_i^f(\xi) \mathbf{f}^i, \end{aligned}$$

where the vector  $\mathbf{u}_n(\xi)$  contains the coefficients of the Galerkin projection onto  $H_n$  expressed in the basis of the snapshots  $(u^\mathcal{N}(\xi^j))_{j=1}^n$ . Since the system size does not depend on  $\mathcal{N}$ , the computational cost for solving these systems (in  $O(n^\kappa)$ ) is much smaller than the computational cost to obtain a truth snapshot (in  $O(\mathcal{N}^\kappa)$  for some  $\kappa > 1$ ). It justifies the interest of such methods when  $\alpha$  and  $\beta$  are of the same order of magnitude. The values of  $\alpha$  and  $\beta$  depend on the nature (sparse, tridiagonal ...) of the linear system to solve.

When the affine decomposition of  $a$  or  $f$  is not obvious, the empirical interpolation method (EIM) [6] provides such a decomposition. It allows then an efficient offline-online decomposition as we have seen before.

### 4.3.3 Error Estimation

There is still one missing piece to propose a practical version of Algorithm 4.1 for parameter-dependent PDEs: the estimation of the error  $\|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}}$ , where  $H_{i-1}$  is the already built reduced space. A common approach in the literature is to propose an estimator based on the residual. We define the residual for all  $v \in \mathcal{H}$  as  $r_{i-1}(v; \xi) := a(u(\xi) - u_{i-1}(\xi), v; \xi)$ . We obtain that

$$r_{i-1}(v; \xi) = f(v; \xi) - a(u_{i-1}(\xi), v; \xi), \quad \forall v \in \mathcal{H}.$$

Then we have the following inequalities

$$(4.24) \quad \beta(\xi) \|e_{i-1}(\xi)\|_{\mathcal{H}} \leq \|r_{i-1}(\cdot; \xi)\|_{\mathcal{H}'(\mathcal{O})} \leq \alpha(\xi) \|e_{i-1}(\xi)\|_{\mathcal{H}},$$

where  $\|r_{i-1}(\cdot; \xi)\|_{\mathcal{H}'(\mathcal{O})} := \sup_{v \in \mathcal{H}} \frac{r_{i-1}(v; \xi)}{\|v\|_{\mathcal{H}}}$  and  $e_{i-1}(\xi) = u(\xi) - u_{i-1}(\xi)$ . The quantity

$$(4.25) \quad \Delta(u_{i-1}(\xi)) := \frac{\|r_{i-1}(\cdot; \xi)\|_{\mathcal{H}'(\mathcal{O})}}{\beta^{\min}}$$

is an upper bound of the true error  $\|e_{i-1}(\xi)\|_{\mathcal{H}}$  and is usually taken as an error estimate. With this choice we have the following result.

**Theorem 4.6.** *We assume that (4.18), (4.19) and (4.24) hold and that the error estimate is as in (4.25). Then Algorithm 4.2 adapted to parameter-dependent PDEs is a  $\gamma$ -weak-greedy algorithm over any training set  $\Xi$  with  $\gamma = \frac{(\beta^{\min})^2}{(\alpha^{\max})^2}$ .*

The use of an error estimate based on the residual degrades the performance of the practical weak-greedy algorithm for parameter-dependent PDEs. It would be more interesting to directly deal with the approximation error  $\|u(\xi) - u_{i-1}(\xi)\|_{\mathcal{H}}$ , particularly for problems with small  $\beta^{\min}$ . We present a new technique to avoid using residual-based error estimate in Chapter 5.



# Chapter 5

## Probabilistic Reduced Basis Methods

In this chapter we present our third contribution. This contribution consists in two probabilistic greedy algorithms for reduced basis methods (RBM). In order to present these algorithms, we first provide a Feynman-Kac representation of the approximation error which allows us interpret its squared norm as an expectation and thus to use Monte-Carlo methods to estimate this squared norm. Then combining this Monte-Carlo estimation with the probably approximately correct (PAC) algorithm from Chapter 3 we come up with a new technique to adaptively select snapshots for the construction of the reduced basis.

### Contents

5.1	Probabilistic Representation for Parameter-Dependent PDEs	94
5.1.1	Elliptic Parameter-Dependent PDEs . . . . .	94
5.1.2	Parabolic Parameter-Dependent PDEs . . . . .	95
5.2	Error Estimation using Probabilistic Representation . . . .	96
5.2.1	Parameter-Dependent Elliptic PDEs . . . . .	97
5.2.2	Parameter-Dependent Parabolic PDEs . . . . .	98
5.2.3	Probably Approximately Correct Algorithm . . . .	99
5.3	Probabilistic Greedy Algorithms . . . . .	100
5.3.1	With Probabilistic Error Estimate . . . . .	100
5.3.2	With Random Training Sets . . . . .	102

We consider, as in Chapter 4, a set of parameters  $Y \subset \mathbb{R}^p$ , e.g.  $Y = [-1, 1]^p$  and the following parameter-dependent problem

$$(5.1) \quad \mathcal{S}(u(\cdot; y); y) = 0$$

where the unknown is a real-valued function  $u(y) := u(\cdot; y)$  belongs to some Hilbert space  $\mathcal{H}$ .

In Chapter 4 traditional deterministic reduced basis methods (RBM) were presented for (5.1). Here we present probabilistic reduced basis methods where probabilistic methods are considered to select the snapshot that enriches the reduced space.

Our problems of interest are parameter-dependent PDEs whose solution  $u(y)$  admits a probabilistic representation, which means that there exists some parameter-dependent functional  $Q_y : \mathbb{R}^d \rightarrow \mathbb{R}$ , depending on the data from (5.1), and some random variable  $X^y$  defined on a probability space  $(\Omega, \mathcal{F}, \mathbb{P})$  such that

$$(5.2) \quad u(y) = \mathbb{E}[Q_y(X^y)].$$

This probabilistic representation of the solution is motivated by the Feynman-Kac representation of solutions of partial differential equations (PDEs). We give in Section 5.1 two guiding examples of such representations for solutions of elliptic and parabolic parameter-dependent PDEs. Those are extensions to parameter-dependent PDEs of the probabilistic representations given in Chapter 1. In Section 5.2 we use this probabilistic representation of the solution to derive a probabilistic representation of the approximation error  $u(y) - u_n(y)$  for any  $y$  in  $Y$ , where  $u_n : Y \rightarrow H_n$  is an approximation of  $u$  onto the finite dimensional (reduced) space  $H_n \subset \mathcal{H}$ . In particular we will reinterpret the  $\mathcal{H}$ -norm of the approximation error as an expectation. It allows to propose a first probabilistic greedy algorithm to construct the reduced space  $H_n$  that is analyzed in Section 5.3.1. Then our second probabilistic greedy algorithm combines the first algorithm with random training sets from [27] which allows us to treat parameter-dependent problems with high-dimensional parametric dimension.

## 5.1 Probabilistic Representation for Parameter-Dependent PDEs

We extend the Feynman-Kac representation from Chapter 1 to parameter-dependent PDEs.

### 5.1.1 Elliptic Parameter-Dependent PDEs

We consider an elliptic parameter-dependent PDE on a open bounded domain  $\mathcal{D} \subset \mathbb{R}^d$ . For any  $y$  in  $Y$ , let

$$(5.3) \quad \begin{aligned} -\mathcal{L}_y u(x; y) &= g(x; y) & x \in \mathcal{D}, \\ u(x; y) &= f(x; y) & x \in \partial\mathcal{D}, \end{aligned}$$

where the partial differential operator is defined as follow

$$\mathcal{L}_y = \frac{1}{2} \sum_{i,j=1}^d (\sigma(x; y) \sigma(x; y)^T)_{ij} \partial_{x_i x_j}^2 + \sum_{i=1}^d b_i(x; y) \partial_{x_i}$$

while  $g : \overline{\mathcal{D}} \times Y \rightarrow \mathbb{R}$  is the source term and  $f : \partial\mathcal{D} \times Y \rightarrow \mathbb{R}$  is the boundary condition. The differential operator  $\mathcal{L}_y$  is also the infinitesimal generator of the stochastic process  $X^{x;y}$  which is solution of the following Stochastic Differential Equation (SDE)

$$(5.4) \quad \begin{aligned} X_t^{x;y} &= X_0^{x;y} + \int_0^t b(s, X_s^{x;y}; y) ds \\ &\quad + \int_0^t \sigma(s, X_s^{x;y}; y) dW_s, \quad X_0^{x;y} = x \in \overline{\mathcal{D}}. \end{aligned}$$

For a fixed value of  $y$  in  $Y$ , we can then state a similar result as Theorem 1.12 that applies to (5.3).

**Theorem 5.1** (Feynman-Kac formula for an elliptic parameter-dependent PDE on bounded domain). *Assume that, for all  $y$  in  $Y$*

(EP1)  $\mathcal{D}$  is an open connected bounded domain of  $\mathbb{R}^d$ , regular in the sense that, if  $\tau^{x;y} = \inf \{s > 0 : X_s^{x;y} \notin \mathcal{D}\}$  is the first exit time of  $\mathcal{D}$  for the process  $X^{x;y}$ , solution of (5.4), we have

$$\mathbb{P}(\tau^{x;y} = 0) = 1, \quad x \in \partial\mathcal{D},$$

(EP2)  $b(\cdot; y), \sigma(\cdot; y)$  are Lipschitz functions over  $\overline{\mathcal{D}}$ ,

(EP3)  $f(\cdot; y)$  is continuous on  $\partial\mathcal{D}$ ,  $g(\cdot; y)$  is Hölder-continuous on  $\overline{\mathcal{D}}$ ,

(EP4) (uniform ellipticity assumption) there exists  $c > 0$  such that

$$\sum_{i,j=1}^d \left( \sigma(x; y) \sigma(x; y)^T \right)_{ij} \Upsilon_i \Upsilon_j \geq c \sum_{i=1}^d \Upsilon_i^2, \quad \Upsilon \in \mathbb{R}^d, x \in \overline{\mathcal{D}}.$$

Then, for all  $y$  in  $Y$ , there exists a unique solution  $u(\cdot; y)$  of (5.3) in  $\mathcal{C}(\overline{\mathcal{D}}) \cap \mathcal{C}^2(\mathcal{D})$ . It satisfies for all  $x \in \overline{\mathcal{D}}$

$$(5.5) \quad u(x; y) = \mathbb{E} \left[ f(X_{\tau^{x;y}}^{x;y}; y) + \int_0^{\tau^{x;y}} g(X_t^{x;y}; y) dt \right].$$

### 5.1.2 Parabolic Parameter-Dependent PDEs

We consider a parabolic parameter-dependent PDE. For any  $y$  in  $Y$

$$(5.6) \quad \begin{aligned} \mathcal{A}_y u(t, x; y) + g(t, x; y) &= 0 \quad (t, x) \in [0, T) \times \mathcal{D}, \\ u(T, x; y) &= f(T, x; y) \quad x \in \overline{\mathcal{D}}, \\ u(t, x; y) &= f(t, x; y) \quad (t, x) \in [0, T) \times \partial\mathcal{D}, \end{aligned}$$

where  $\mathcal{A}_y = \partial_t + \mathcal{L}_y$ , with the differential operator  $\mathcal{L}_y$  is defined as follow

$$\mathcal{L}_y = \frac{1}{2} \sum_{i,j=1}^d (\sigma(t, x; y) \sigma(t, x; y)^T)_{ij} \partial_{x_i x_j}^2 + \sum_{i=1}^d b_i(t, x; y) \partial_{x_i},$$

while  $g : [0, T] \times \overline{\mathcal{D}} \times Y \rightarrow \mathbb{R}$  is the source term and  $f : [0, T] \times \overline{\mathcal{D}} \times Y \rightarrow \mathbb{R}$  represents both the terminal and boundary condition. In the parabolic case,  $\mathcal{L}_y$  can also be seen as the infinitesimal generator of the stochastic process  $X^{t,x;y}$  which is solution of the following Stochastic Differential Equation (SDE)

$$(5.7) \quad \begin{aligned} X_s^{t,x;y} &= X_t^{t,x;y} + \int_t^s b(r, X_r^{t,x;y}; y) dr \\ &\quad + \int_t^s \sigma(r, X_r^{t,x;y}; y) dW_r, \quad X_t^{t,x;y} = x \in \overline{\mathcal{D}}. \end{aligned}$$



**Theorem 5.2** (Feynman-Kac formula for a backward parabolic parameter-dependent PDE on bounded domain). *Let  $T > 0$  and assume that, for all  $y$  in  $Y$*

(PP1) *assumptions (A1) and (A2) are satisfied for  $b(\cdot; y)$  and  $\sigma(\cdot; y)$ ,*

(PP2)  *$\mathcal{D} \subset \mathbb{R}^d$  is a bounded domain,*

(PP3) *the boundary  $\partial\mathcal{D}$  is regular for  $X^{t,x;y}$  in the sense that*

$$(5.8) \quad \forall t \in [0, T], \forall x \in \partial\mathcal{D}, \mathbb{P}(\tau^{t,x;y} = 0) = 1,$$

*with  $\tau^{t,x;y} = \inf\{s > t : X_s^{t,x;y} \notin \mathcal{D}\}$  the first exit time of  $X^{t,x;y}$  from  $\mathcal{D}$ ,*

(PP4) *the functions  $f(\cdot; y), g(\cdot; y) : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$  are continuous,*

(PP5) *there exists a continuous function  $u(\cdot; y) : [0, T] \times \overline{\mathcal{D}} \rightarrow \mathbb{R}$ , of class  $\mathcal{C}^{1,2}$  on all open subsets of  $[0, T) \times \mathcal{D}$  verifying (5.6).*

*Then  $u(\cdot; y)$  admits the following probabilistic representation*

$$(5.9) \quad u(t, x; y) = \mathbb{E} \left[ f(\tau_T^{t,x;y}, X_{\tau_T^{t,x;y}}^{t,x;y}; y) + \int_t^{\tau_T^{t,x;y}} g(s, X_s^{t,x;y}; y) ds \right]$$

*where  $\tau_T^{t,x;y} = \tau^{t,x;y} \wedge T$ .*

## 5.2 Error Estimation using Probabilistic Representation

In the context of RBM, a greedy algorithm consists, at each step  $i + 1 \geq 1$ , in seeking  $y$  that maximizes the approximation error to enrich the current reduced space  $H_i \subset \mathcal{H}$ . This optimization problem usually takes the following form

$$(5.10) \quad \sup_{y \in Y} \|u(y) - u_i(y)\|_{\mathcal{H}},$$

where  $u_i(y) : \mathcal{O} \rightarrow \mathbb{R}$  is some approximation of  $u(y)$  in some space  $H_i$  (e.g. obtained by interpolation, least-squares or Galerkin projection). We assume all along this section that both the approximation  $u_i$  and the space  $H_i$  are deterministic. In this section, we propose to interpret the squared norm of the approximation error  $\|u(y) - u_i(y)\|_{\mathcal{H}}^2$  as an expectation using probabilistic representations introduced in Section 5.1. We assume here that, for any  $y$  in  $Y$ , we have

$$(5.11) \quad u_i(y) = \sum_{j=1}^i \alpha_j(y) \phi_j,$$

where the family of functions  $(\phi_j)_{j=1 \dots i}$  is a basis of  $H_i$ , obtained from snapshots  $u(y^j)$  selected for example by a greedy algorithm.

### 5.2.1 Parameter-Dependent Elliptic PDEs

Assuming that  $u_i(y)$  is regular enough (see Theorem 5.3) for all  $y$  in  $Y$  (through the choice of the reduced basis functions  $\phi_j$ ), we have that  $e_i(y) := u(y) - u_i(y)$  is the unique solution, for all  $y$  in  $Y$ , of

$$(5.12) \quad \begin{aligned} -\mathcal{L}_y e_i(y) &= g_i(y) \quad \text{on } \mathcal{D}, \\ e_i(y) &= f_i(y) \quad \text{on } \partial\mathcal{D}, \end{aligned}$$

where  $f_i(y) := f(y) - u_i(y)$  and  $g_i(y) = g(y) + \mathcal{L}_y u_i(y)$ . Applying Theorem 5.1, we obtain the following probabilistic representation for  $e_i(y)$  for any  $y$  in  $Y$ :

**Theorem 5.3** (Feynman-Kac formula for  $e_i(y) = u(y) - u_i(y)$  for elliptic parameter-dependent PDEs). *Assume that, for all  $y$  in  $Y$ , (EP1)- (EP3) hold and  $u_i(y)$  is of class  $\mathcal{C}(\overline{\mathcal{D}}) \cap \mathcal{C}^2(\mathcal{D})$  such that  $\mathcal{L}_y u_i(y)$  is Hölder-continuous on  $\overline{\mathcal{D}}$ . Then, for all  $y$  in  $Y$ ,  $e_i(y)$  is the unique solution of (5.12) in  $\mathcal{C}(\overline{\mathcal{D}}) \cap \mathcal{C}^2(\mathcal{D})$  and satisfies for all  $x \in \overline{\mathcal{D}}$*

$$(5.13) \quad \begin{aligned} e_i(x; y) &= \mathbb{E} [F_i(x, X^{x;y}; y)] \\ &:= \mathbb{E} \left[ f_i(X_{\tau^{x;y}}^{x;y}; y) + \int_0^{\tau^{x;y}} g_i(X_t^{x;y}; y) dt \right]. \end{aligned}$$

In what follows we suppose that  $\mathcal{H} \subset L^2(\mathcal{D})$  is equipped with the  $L^2$ -norm over  $\mathcal{D}$ . Using the probabilistic representation of  $e_i(y)$ , we prove the following reinterpretation of  $\|e_i(y)\|_{L^2}^2$ .

**Theorem 5.4** (Reinterpretation of  $\|e_i(y)\|_{L^2}^2$ ). *Taking  $V \sim \mathcal{U}(\mathcal{D})$  uniformly distributed on  $\mathcal{D}$ , we have for any  $y$  in  $Y$*

$$(5.14) \quad \|e_i(y)\|_{L^2}^2 = |\mathcal{D}| \mathbb{E} [F_i(V, X^{V;y}; y) F_i(V, \tilde{X}^{V;y}; y)],$$

where  $X^{x;y}$  and  $\tilde{X}^{x;y}$  are two i.i.d random processes for any  $x \in \overline{\mathcal{D}}$  and  $|\mathcal{D}|$  is the Lebesgue measure of  $\mathcal{D}$ .

*Proof.* We first recall

$$\begin{aligned} \|e_i(y)\|_{L^2}^2 &= \int_{\mathcal{D}} (e_i(x; y))^2 dx \\ &= |\mathcal{D}| \mathbb{E} [(e_i(V; y))^2]. \end{aligned}$$

Then, by the tower property

$$\begin{aligned} \mathbb{E} [(e_i(V; y))^2] &= \mathbb{E} [\mathbb{E} [F_i(V, X^{V;y}; y) | V]^2] \\ &= \mathbb{E} [\mathbb{E} [F_i(V, X^{V;y}; y) | V] \mathbb{E} [F_i(V, \tilde{X}^{V;y}; y) | V]]. \end{aligned}$$

Then, by independence of  $X^{V;y} | V$  and  $\tilde{X}^{V;y} | V$  we can write

$$\mathbb{E} [(e_i(V; y))^2] = \mathbb{E} [\mathbb{E} [F_i(V, X^{V;y}; y) F_i(V, \tilde{X}^{V;y}; y) | V]].$$

Finally, by the law of iterated expectation, we get (5.14).  $\square$

**Remark 5.5.** Assuming the existence of probabilistic representations for the gradient of  $u(y)$  and  $u_i(y)$ , it would be possible to consider probabilistic interpretation of other squared norms of the approximation error, such as the  $H^1$ -norm. Such probabilistic representations have been derived in simple cases, see e.g. [47, Corollary IV.5.2].

### 5.2.2 Parameter-Dependent Parabolic PDEs

Here the functions  $u(y)$ ,  $u_i(y)$ ,  $f(y)$ ,  $g(y)$  and  $e_i(y)$  depend on  $t$  and  $x$ , which we will not precise when there is no ambiguity. In this section, the  $L^2$ -norm is taken over  $[0, T] \times \mathcal{D}$ .

Assuming that  $u_i(y)$  is regular enough (see Theorem 5.6) for all  $y$  in  $Y$  (through the choice of the reduced basis functions  $\phi_i$ ), we have that the approximation error  $e_i(y) := u(y) - u_i(y)$  satisfies, for all  $y$  in  $Y$ , the following equations

$$(5.15) \quad \begin{aligned} \mathcal{A}_y e_i(y) + g_i(y) &= 0 \quad \text{on} \quad [0, T] \times \mathcal{D}, \\ e_i(y) &= f_i(y) \quad \text{on} \quad \{T\} \times \overline{\mathcal{D}}, \\ e_i(y) &= f_i(y) \quad \text{on} \quad [0, T] \times \partial\mathcal{D}, \end{aligned}$$

where  $f_i(y) := f(y) - u_i(y)$  and  $g_i(y) = g(y) + \mathcal{A}_y u_i(y)$ . Applying Theorem 5.2, we obtain the following probabilistic representation for the unknown  $e_i(y) = u(y) - u_i(y)$ .

**Theorem 5.6** (Feynman-Kac formula for  $e_i(y) = u(y) - u_i(y)$  for parabolic parameter-dependent PDEs). Assume that, for all  $y$  in  $Y$ , (PP1)-(PP5) hold and assume moreover that  $u_i(y)$  is continuous on  $[0, T] \times \overline{\mathcal{D}}$ , of class  $\mathcal{C}^{1,2}$  on all open sets of  $[0, T] \times \mathcal{D}$  and such that  $\mathcal{A}_y u_i$  is continuous on  $[0, T] \times \overline{\mathcal{D}}$ .

Then, for all  $y$  in  $Y$ ,  $e_i(\cdot, y)$  admits the following probabilistic representation

$$(5.16) \quad \begin{aligned} e_i(t, x; y) &= \mathbb{E}[F_i(t, x, X^{t,x;y}; y)] \\ &:= \mathbb{E} \left[ f_i(\tau^{x;y}, X_{\tau^{x;y}}^{t,x;y}; y) + \int_t^{\tau^{x;y}} g_i(s, X_s^{t,x;y}; y) ds \right]. \end{aligned}$$

Using the above probabilistic representation, we prove the following expression of the squared  $L^2$ -norm of the approximation error, following the proof of Theorem 5.4.

**Theorem 5.7** (Reinterpretation of  $\|e_i(y)\|_{L^2}^2$ ). Taking  $V \sim \mathcal{U}(\mathcal{D})$  uniformly distributed on  $\mathcal{D}$  and  $\mathbf{T} \sim \mathcal{U}(0, T)$  uniformly distributed on  $[0, T]$ , we have for any  $y$  in  $Y$

$$\|e_i(y)\|_{L^2}^2 = T|\mathcal{D}| \mathbb{E} \left[ F_i(\mathbf{T}, V, X^{\mathbf{T},V;y}; y) F_i(\mathbf{T}, V, \tilde{X}^{\mathbf{T},V;y}; y) \right],$$

where  $X^{t,x;y}$  and  $\tilde{X}^{t,x;y}$  are two i.i.d random processes for any  $(t, x) \in [0, T] \times \overline{\mathcal{D}}$ .

### 5.2.3 Probably Approximately Correct Algorithm

As explained in Chapter 4, solving the global optimization problem (5.10) is not realistic when  $Y$  is not a discrete set. We rather consider the following discretized optimization problem

$$(5.17) \quad \max_{\xi \in \Xi} \|u(\xi) - u_i(\xi)\|_{\mathcal{H}},$$

where  $\Xi \subset Y$  is a finite set of parameters. Thus, using Theorem 5.4 or Theorem 5.7 when  $\|\cdot\|_{\mathcal{H}} = \|\cdot\|_{L^2}$ , (5.17) becomes a discrete optimization problem where the quantity to optimize is the expectation of some parameter-dependent random variable.

We describe the procedure to sample  $X^{W;\xi}$  (with  $W = V$  for the elliptic case or  $W = (\mathbf{T}, V)$  for the parabolic case):

1. draw a sample  $w$  according to the distribution of  $W$  ;
2. simulate using a time-integration scheme an approximate trajectory of the diffusion process  $X^{w;\xi}$  that will be denoted  $X^{\Delta t, w; \xi}$ , see (1.15).

These samples are used to provide Monte-Carlo estimates of  $\|u(\xi) - u_i(\xi)\|_{L^2}^2$  for different values of  $\xi$  in  $\Xi$ . However our goal is not to provide precise estimates for  $\|u(\xi) - u_i(\xi)\|_{L^2}^2$  for every value  $\xi$  in  $\Xi$ , which would require a high number of samples from  $X^{\Delta t, W; \xi}$ . We rather use the PAC algorithm from Chapter 3 to solve (5.17) in high probability with a guaranteed relative quasi-optimality, sampling less the random variables  $Z_{i+1}(\xi) = |\mathcal{O}| F_i(W, X^{W;\xi}; \xi) F_i(W, \tilde{X}^{W;\xi}; \xi)$  ( $|\mathcal{O}|$  is Lebesgue measure of the physical domain  $\mathcal{O}$  and is equal to  $|\mathcal{D}|$  in the elliptic case and to  $T|\mathcal{D}|$  in the parabolic case). This algorithm returns  $\xi^{i+1}$  in  $\Xi$  such that

$$(5.18) \quad \mathbb{P} \left( \mathbb{E}[Z_{i+1}(\xi^{i+1, \star})] - \mathbb{E}[Z_{i+1}(\xi^{i+1})] \leq \epsilon \mathbb{E}[Z_{i+1}(\xi^{i+1, \star})] \right) \geq 1 - \lambda,$$

for any prescribed threshold  $\epsilon$  in  $(0, 1)$  and probability of failure  $\lambda$  in  $(0, 1)$ , where  $\xi^{i+1, \star} \in \arg \max_{\xi \in \Xi} \mathbb{E}[Z_{i+1}(\xi)]$ .

**Remark 5.8.** For parabolic parameter-dependent PDEs, the random variables  $Z_{i+1}(\xi)$  are bounded for all  $\xi$  in  $\Xi$  since  $\tau_1^{t, x; \xi} \leq T$ , see (5.9). This is not the case for elliptic parameter-dependent PDEs since we usually cannot provide a deterministic bound for the exit time  $\tau^{x; \xi}$  (see (5.5)).

**Remark 5.9.** We chose here not to consider

$$Z_{i+1}^{\Delta t}(\xi) = |\mathcal{O}| F_i(W, X^{\Delta t, W; \xi}; \xi) F_i(W, \tilde{X}^{\Delta t, W; \xi}; \xi)$$

instead of  $Z_{i+1}(\xi)$  in order to avoid dealing with the time-integration error.

### 5.3 Probabilistic Greedy Algorithms

We present in Section 5.3.1 a first greedy algorithm using probabilistic error estimate. Then we recall in Section 5.3.2 another greedy algorithm from [27] using random training sets. Then these algorithms are combined in Algorithm 5.3.

A similar contribution can be found in [16], where the authors proposed a variance reduction algorithm using the reduced basis paradigm for parameter-dependent stochastic differential equations. The analysis of this algorithm can be found in [14].

One has to be careful in this section, since neither the approximation  $u_i$  nor the reduced space  $H_i$  are deterministic. They depend on the previous (probabilistic) steps of the greedy algorithms. We have thus to make a distinction between what is simulated before stage  $i$  and at stage  $i$  in order to analyse these algorithms. We will define appropriate conditional expectations before presenting each one of them.

In this section, we consider the  $L^2$ -norm in order to reuse the probabilistic interpretation of the error proposed in Section 5.2.

#### 5.3.1 With Probabilistic Error Estimate

We describe a first probabilistic greedy algorithm that selects randomly the snapshot to enrich the reduced space using the probabilistic representation of the error introduced in Section 5.2.

---

##### Algorithm 5.1 Probabilistic Greedy Algorithm for RBM

---

**Require:**  $(\lambda_i)_{1 \leq i \leq n} \in (0, 1)$ ,  $\epsilon \in (0, 1)$ ,  $\Xi$  a discrete training set.

- 1: Set  $H_0 = \{0\}$ .
  - 2: **for**  $i = 1 \dots n$  **do**
  - 3:   Select  $\xi^i$  using PAC algorithm to solve  $\max_{\xi \in \Xi} \mathbb{E}[Z_i(\xi)]$  with relative precision  $\epsilon$  and probability greater than  $1 - \lambda_i$ .
  - 4:   Compute  $u(\xi^i)$  and update  $H_i = \text{span}\{u(\xi^1), \dots, u(\xi^i)\}$ .
  - 5: **end for**
- 

**Theorem 5.10.** *Take  $(\lambda_i)_{1 \leq i \leq n} \in (0, 1)$  such that  $\sum_{i=1}^n \lambda_i < 1$ ,  $\epsilon \in (0, 1)$  and  $\Xi$  a discrete training set. Assume moreover that for all  $i = 1, \dots, n$*

$$(5.19) \quad \|u(y) - u_{i-1}(y)\|_{L^2} \leq C \|u(y) - P_{H_{i-1}} u(y)\|_{L^2},$$

*for some constant  $C$  independent from  $H_{i-1}$  and  $y$ . Then, with probability at least  $1 - \sum_{i=1}^n \lambda_i$ , Algorithm 5.1 is a weak-greedy algorithm of parameter  $\frac{\sqrt{1-\epsilon}}{C}$  over  $\Xi$ , which means that for all  $i = 1, \dots, n$*

$$(5.20) \quad \|u(\xi^i) - P_{H_{i-1}} u(\xi^i)\|_{L^2} \geq \frac{\sqrt{1-\epsilon}}{C} \max_{\xi \in \Xi} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{L^2}.$$

*Proof.* In order to analyse Algorithm 5.1, we define here the following conditional expectation, for any  $i \geq 1$  and any random variable on  $(\Omega, \mathcal{F}, \mathbb{P})$

$$(5.21) \quad \mathbb{E}^i[\cdot] = \mathbb{E}[\cdot | Z^{<i}],$$

where  $Z^{<i}$  is the random variables  $Z_{j,k}(\xi)$  for  $\xi$  in  $\Xi$  and  $j < i$ , where  $Z_{j,k}(\xi)$  are i.i.d. copies of  $Z_j(\xi)$  generated by PAC algorithm at step  $j$  of the greedy algorithm. Then for any element  $A$  in  $\mathcal{F}$ , we let

$$(5.22) \quad \mathbb{P}^i(A) = \mathbb{E}^i[\mathbb{1}_A].$$

Using these notations we can write, in the case where  $u_i$  is randomly determined by the steps before  $i$ , that

$$(5.23) \quad \|e_i(y)\|_{L^2}^2 = \|u(y) - u_i(y)\|_{L^2}^2 = \mathbb{E}^{i+1}[Z_{i+1}(\xi)].$$

These notations being introduced, we define the following event  $A = \cap_{i=1}^n A_i$ , each  $A_i$  being defined as

$$A_i := \left\{ \mathbb{E}^i[Z_i(\xi^{i,*})] - \mathbb{E}^i[Z_i(\xi^i)] \leq \epsilon \mathbb{E}^i[Z_i(\xi^{i,*})] \right\}.$$

We have  $\mathbb{P}^i(A_i) \geq 1 - \lambda_i$  almost surely. Now, for all  $\omega \in \Omega$  such that  $A$  is verified, we have for any  $i \geq 1$

$$\begin{aligned} \|u(\xi^i) - P_{H_{i-1}}u(\xi^i)\|_{L^2}^2 &\geq \frac{1}{C^2} \|u(\xi^i) - u_{i-1}(\xi^i)\|_{L^2}^2 \\ &= \frac{1}{C^2} \mathbb{E}^i[Z_i(\xi^i)] \\ &\geq \frac{1-\epsilon}{C^2} \mathbb{E}^i[Z_i(\xi^{i,*})] \\ &= \frac{1-\epsilon}{C^2} \max_{\xi \in \Xi} \mathbb{E}^i[Z_i(\xi)] \\ &= \frac{1-\epsilon}{C^2} \max_{\xi \in \Xi} \|u(\xi) - u_{i-1}(\xi)\|_{L^2}^2 \\ &\geq \frac{1-\epsilon}{C^2} \max_{\xi \in \Xi} \|u(\xi) - P_{H_{i-1}}u(\xi)\|_{L^2}^2 \end{aligned}$$

Taking the square root of the obtained inequality, we obtain (5.20). We now estimate  $\mathbb{P}(A)$

$$\begin{aligned} \mathbb{P}(A) &= 1 - \mathbb{P}(\overline{A}) \geq 1 - \sum_{i=1}^n \mathbb{P}(\overline{A_i}) \\ &= 1 - \sum_{i=1}^n \mathbb{E}[\mathbb{1}_{\overline{A_i}}] \\ &= 1 - \sum_{i=1}^n \underbrace{\mathbb{E}[\mathbb{E}[\mathbb{1}_{\overline{A_i}} | Z^{<i}]]}_{\mathbb{P}^i(\overline{A_i})} \\ &\geq 1 - \sum_{i=1}^n \lambda_i, \end{aligned}$$

where the last inequality derives from the selection procedure of  $\xi^i$ , see step 3 of Algorithm 5.1. It gives us the desired result.  $\square$

### 5.3.2 With Random Training Sets

We consider now a Hilbert space  $\mathcal{H}$ . We have not mentioned how we choose  $\Xi$  so far. For low-dimensional parameter spaces  $Y \subset \mathbb{R}^p$ , i.e. when  $p$  is relatively small, it is possible to take for  $\Xi$  some  $\epsilon$ -net of  $Y$  with a small  $\epsilon$ . However when  $p$  is too high, this choice for  $\Xi$  is not reasonable since the number of parameters in  $\Xi$  grows exponentially with  $p$ . The authors in [27] proposed a solution when  $p$  is large, by using random training sets. We recall in Algorithm 5.2 the greedy algorithm using these random training sets whose analysis is made in Theorem 5.11.

---

**Algorithm 5.2** Greedy Algorithm with Random Training Sets [27]

---

**Require:**  $(\Xi_i)_{i=1\dots m^2}$  a collection of discrete training sets.

- 1: Set  $i = 0$ ,  $H_0 = \{0\}$  and  $\hat{\sigma}_0 = \frac{\epsilon}{8m}$ .
- 2: **while**  $\hat{\sigma}_i \geq \frac{\epsilon}{8m}$  and  $i < m^2$  **do**
- 3:   Increment  $i \leftarrow i + 1$ .
- 4:   Select  $\xi^i \in \Xi_i$  such that

$$\xi^i \in \arg \max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}},$$

$$\text{and } \hat{\sigma}_i = \max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}}.$$

- 5:   Compute  $u(\xi^i)$  and update  $H_i = \text{span}\{u(\xi^1), \dots, u(\xi^i)\}$ .
  - 6: **end while**
- 

Before presenting the analysis of this algorithm, we recall a few notations. We introduce the polynomial space  $\mathcal{P}_\Lambda := \text{span}\{y \mapsto y^\nu := \prod_{j=1}^p y_j^{\nu_j} : \nu \in \Lambda\}$  where  $\nu$  is a multi-index and  $\Lambda$  a downward closed set of multi-indices. Then  $\mathcal{Q}_\Lambda := \mathcal{H} \otimes \mathcal{P}_\Lambda$  is the set of functions from  $Y$  to  $\mathcal{H}$  that can be written  $u(y) = \sum_{\nu \in \Lambda} c_\nu y^\nu$  with  $c_\nu \in \mathcal{H}$ . We also define  $\Sigma_m := \cup_{\#\Lambda=m} \mathcal{Q}_\Lambda$  the set of functions from  $Y$  to  $\mathcal{H}$  that admits a  $m$ -term expansion on the monomial basis  $\{y^\nu : \nu \in \mathbb{N}^p\}$ . With these notations, the class of functions  $\mathcal{A}^r := \mathcal{A}^r((\Sigma_m)_{m \geq 1})$  is composed of all the functions  $v : Y \rightarrow \mathcal{H}$  in  $L^\infty(Y, \mathcal{H})$  such that

$$(5.24) \quad \delta_m(u) := \inf_{P \in \Sigma_m} \sup_{y \in Y} \|u(y) - P(y)\|_{\mathcal{H}} \leq C m^{-r}, \quad m \geq 1.$$

The norm  $\|\cdot\|_{\mathcal{A}^r}$  can be defined on  $\mathcal{A}^r$  by  $\|v\|_{\mathcal{A}^r} := \max\{\|v\|_{L^\infty(Y, \mathcal{H})}, |v|_{\mathcal{A}^r}\}$ , where  $|v|_{\mathcal{A}^r}$  is defined as the smallest  $C$  such that  $\delta_m(v) \leq C m^{-r}$ .

**Theorem 5.11.** [27, Theorem 2] Take  $\eta, \epsilon$  in  $(0, 1)$ . Assume that

- $y \mapsto u(y)$  belongs to  $\mathcal{A}^r$  for some  $r > 2$  ;
- $\|u\|_{\mathcal{A}^r} \leq M_0$  for some  $M_0 > 0$  ;
- $m$  is the smallest integer such that  $32M_0 m^{-r+2} \leq \epsilon$  and  $2^{4r+2} m^{-r} \leq 1$  ;
- $N$  is the smallest integer such that  $(1 - \frac{3}{4m^2})^N \leq \frac{\eta}{m^2}$  ;
- $(\Xi_i)_{i=1\dots m^2} \subset Y$  is a collection of discrete training sets obtained from  $N$  samples drawn according to the uniform distribution over  $Y = [-1, 1]^d$ .

Then, with probability greater than  $1 - \eta$ , Algorithm 5.2 is a  $\gamma$ -weak-greedy algorithm with  $\gamma = 1/8m$  over  $Y$ , which means that at each step before the stopping of the algorithm

$$(5.25) \quad \|u(\xi^i) - P_{H_{i-1}}u(\xi^i)\|_{\mathcal{H}} \geq \gamma \sup_{y \in Y} \|u(y) - P_{H_{i-1}}u(y)\|_{\mathcal{H}}.$$

Moreover, denoting by  $H_f$  the reduced space built at the end of Algorithm 5.2, we have  $\text{dist}(\mathcal{M}, H_f) := \sup_{y \in Y} \|u(y) - P_{H_f}u(y)\|_{\mathcal{H}} \leq \epsilon$  with probability greater than  $1 - \eta$ .

*Proof.* We only repeat here the proof of [27, Theorem 4.1] for the first part of the proof, with some details useful for the next results.

We define the following event  $B = \cap_{i=1}^f B_i$  where  $f$  is when Algorithm 5.2 stops and  $B_i$  is defined by

$$B_i = \{\hat{\sigma}_i \geq \gamma \sigma_i(\mathcal{M})_{\mathcal{H}}\},$$

where  $\sigma_i(\mathcal{M})_{\mathcal{H}} = \text{dist}(\mathcal{M}, H_{i-1})_{\mathcal{H}}$ . We recall  $\hat{\sigma}_i = \max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}}u(\xi)\|_{\mathcal{H}}$ . We observe  $B \supset \cap_{i=1}^{m^2} B_i$ . Thus we can write

$$\begin{aligned} \mathbb{P}(B) &= 1 - \mathbb{P}(\overline{B}) \geq 1 - \sum_{i=1}^{m^2} \mathbb{P}(\overline{B}_i) \\ &= 1 - \sum_{i=1}^{m^2} \mathbb{E} [\mathbb{1}_{\overline{B}_i}] \\ &= 1 - \sum_{i=1}^{m^2} \mathbb{E} \left[ \mathbb{E} [\mathbb{1}_{\overline{B}_i} | \Xi_1, \dots, \Xi_{i-1}] \right]. \end{aligned}$$

For fixed training sets  $\Xi_1, \dots, \Xi_{i-1}$ ,  $H_{i-1}$  is fixed and [27, Lemma 3.1] gives

$$\mathbb{E} [\mathbb{1}_{\overline{B}_i} | \Xi_1, \dots, \Xi_{i-1}] = \mathbb{P}(\overline{B}_i | \Xi_1, \dots, \Xi_{i-1}) \leq \frac{\eta}{m^2},$$

which concludes our proof.  $\square$

Algorithm 5.1 and Algorithm 5.2 can be combined to give Algorithm 5.3 that we analyze in Theorem 5.12. In order to do so, we take  $\mathcal{H} = L^2(\mathcal{O})$  and suppose that we consider only the  $L^2$ -norm.

---

**Algorithm 5.3** Full Probabilistic Greedy Algorithm
 

---

**Require:**  $(\lambda_i)_{1 \leq i \leq m^2} \in (0, 1)$ ,  $(\Xi_i)_{i=1 \dots m^2}$  a collection of discrete training sets.

- 1: Set  $i = 0$ ,  $H_0 = \{0\}$  and  $\hat{\sigma}_0 = \frac{\epsilon}{8m}$ .
  - 2: **while**  $\hat{\sigma}_i \geq \frac{\epsilon}{8m}$  and  $i < m^2$  **do**
  - 3:   Increment  $i \leftarrow i + 1$ .
  - 4:   Select  $\xi^i$  using PAC algorithm to solve  $\max_{\xi \in \Xi_i} \mathbb{E}[Z_i(\xi)]$  with relative precision  $\epsilon$  and probability greater than  $1 - \lambda_i$ .
  - 5:   Set  $\hat{\sigma}_i = \max_{\xi \in \Xi_i} \sqrt{\mathbb{E}[Z_i(\xi)]}$ .
  - 6:   Compute  $u(\xi^i)$  and update  $H_i = \text{span}\{u(\xi^1), \dots, u(\xi^i)\}$ .
  - 7: **end while**
-



**Theorem 5.12.** *Under assumptions of Theorem 5.10 and Theorem 5.11, we have that Algorithm 5.3 is a  $\gamma$ -weak-greedy algorithm over  $Y$  with  $\gamma = \frac{\sqrt{1-\epsilon}}{8mC}$ , and with probability greater than  $1 - \eta - \sum_{i=1}^{m^2} \lambda_i$ .*

*Proof.* In order to analyse Algorithm 5.3, we define here the following conditional expectation, for any  $i \geq 1$  and any random variable on  $(\Omega, \mathcal{F}, \mathbb{P})$

$$(5.26) \quad \mathbb{E}^i[\cdot] = \mathbb{E}[\cdot | Z^{<i}],$$

where  $Z^{<i}$  is the  $\Xi_j$  for  $j < i$  and the random variables  $Z_{j,k}(\xi)$  for  $\xi$  in  $\Xi_j$  and  $j < i$ , where  $Z_{k,j}(\xi)$  are i.i.d. copies of  $Z_j(\xi)$  generated by PAC algorithm at step  $j$  of the greedy algorithm. Using these notations we can write, in the case where  $u_i$  is randomly determined by the steps before  $i$ , that

$$(5.27) \quad \|e_i(y)\|_{L^2}^2 = \|u(y) - u_i(y)\|_{L^2}^2 = \mathbb{E}^{i+1}[Z_{i+1}(\xi)].$$

Taking  $f$  the final step of the algorithm, let  $A = \cap_{i=1}^f A_i$  and  $B = \cap_{i=1}^f B_i$  with

$$\begin{aligned} A_i &:= \left\{ \mathbb{E}^i[Z_i(\xi^{i,*})] - \mathbb{E}^i[Z_i(\xi^i)] \leq \epsilon \mathbb{E}^i[Z_i(\xi^{i,*})] \right\}, \\ B_i &= \{ \hat{\sigma}_i \geq \gamma \sigma_i(\mathcal{M})_{L^2} \}, \end{aligned}$$

where  $\sigma_i(\mathcal{M})_{\mathcal{H}} = \text{dist}(\mathcal{M}, H_{i-1})_{\mathcal{H}}$ . We recall  $\hat{\sigma}_i = \max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{\mathcal{H}}$ . We first observe, that for all  $\omega$  in  $\Omega$  such that  $A \cap B$  is verified, we have for any  $i$

$$\begin{aligned} \|u(\xi^i) - P_{H_{i-1}} u(\xi^i)\|_{L^2} &\geq \frac{\sqrt{1-\epsilon}}{C} \max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{L^2} = \frac{\sqrt{1-\epsilon}}{C} \hat{\sigma}_i \\ &\geq \frac{\sqrt{1-\epsilon}}{8mC} \sigma_i(\mathcal{M})_{L^2} = \frac{\sqrt{1-\epsilon}}{8mC} \max_{y \in Y} \|u(y) - P_{H_{i-1}} u(y)\|_{L^2}. \end{aligned}$$

Using  $A \supset \cap_{i=1}^{m^2} A_i$  and  $B \supset \cap_{i=1}^{m^2} B_i$ , we can write

$$\begin{aligned} \mathbb{P}(A \cap B) &= 1 - \mathbb{P}(\overline{A} \cup \overline{B}) \geq 1 - \sum_{i=1}^{m^2} \mathbb{P}(\overline{A}_i) - \sum_{i=1}^{m^2} \mathbb{P}(\overline{B}_i) \\ &= 1 - \sum_{i=1}^{m^2} \mathbb{E}[\mathbb{1}_{\overline{A}_i}] - \sum_{i=1}^{m^2} \mathbb{E}[\mathbb{1}_{\overline{B}_i}] \\ &= 1 - \sum_{i=1}^{m^2} \mathbb{E}[\mathbb{E}[\mathbb{1}_{\overline{A}_i} | Z^{<i}, \Xi_i]] \\ &\quad - \sum_{i=1}^{m^2} \mathbb{E}[\mathbb{E}[\mathbb{1}_{\overline{B}_i} | Z^{<i}]]. \end{aligned}$$

Then we proceed as in the proofs of Theorem 5.10 and Theorem 5.11 to conclude the proof.  $\square$

**Remark 5.13.** *We keep the same stopping criterion as in Algorithm 5.2 for Algorithm 5.3. However this stopping criterion would not be available for a practical*

implementation of Algorithm 5.3 since we cannot compute or give any precise estimation of  $\max_{\xi \in \Xi_i} \|u(\xi) - P_{H_{i-1}} u(\xi)\|_{L^2} = \sqrt{\max_{\xi \in \Xi_i} \mathbb{E}[Z_i(\xi)]}$ . However, e.g. using the bounds of the confidence intervals provided for the estimates of the expectations by PAC algorithm, it is possible to give a practical stopping criterion that can be used for Algorithm 5.3 while ensuring the reaching of a given precision.



# Conclusion and Outlooks

In Chapter 1 we have surveyed standard probabilistic methods for partial differential equations (PDEs), including the probabilistic representation for the solution of a PDE and Monte-Carlo methods coupled with time-integration techniques to get pointwise estimations of this solution. Then, in Chapter 2, we have proposed two algorithms to approximate the global solution of high-dimensional PDEs whose solution admits a probabilistic interpretation. These algorithms combine an adaptive sparse interpolation method from [22] with a control variates technique from [49]. Our first algorithm uses the adaptive sparse interpolation method from [22] to build adaptively the control variate from [49]. This first algorithm is not very stable and the approximation it builds is not precise compared to the one resulting from our second algorithm. This second algorithm is a perturbed version of the adaptive sparse interpolation method from [22]. At each step of the algorithm, we compute the approximation resulting from the control variates technique from [49] on the interpolation basis rather than the interpolation of the exact solution on this interpolation basis. Both algorithms were illustrated theoretically and numerically. We have in particular showed how the second algorithm does not only allow to get precise estimations of the solution on a fixed interpolation grid, such as it has been done in [49], but on the entire domain of definition of the solution. An approximation up to machine precision was built by the second algorithm for an elliptic PDE in dimension 20.

Other adaptive approximation techniques, such as adaptive least-squares, can be used instead of adaptive interpolation techniques. Together with a precise study of the time-integration error for some functional of a stochastic process, it might lead to more advanced results of the study of the convergence of our algorithms.

Then we have studied in Chapter 3 an optimization problem over a discrete set of random variables where the quantity to optimize is their expectation. We have combined expectation estimation in relative precision [79] with a probably approximately correct algorithm in absolute precision [38] to propose two probably approximately correct algorithms in relative precision. The first one naively builds precise estimates for each expectation to finally select the random variable with the highest estimate. The second one adaptively samples each random variable and relies on confidence intervals for these estimates to decide which random variable to continue sampling and which random variable to stop sampling. Both algorithms provide the same theoretical guarantees for their output. However the second algorithm allows important savings in terms of number of samples used compared to the first

algorithm.

These discrete optimization techniques could have been considered as remote from the original interest of this thesis, which is partial differential equations. However, such optimization techniques could be adapted to find extrema of the solution of a PDE. Indeed, when a solution admits a probabilistic representation, it can be written as an expectation of some random process, as we have seen in Chapter 1.

Moreover this probably approximately correct algorithm in relative precision can be used in reduced basis methods. We have surveyed first in Chapter 4 standard reduced basis methods with a particular focus on parameter-dependent PDEs. Then we have proposed two probabilistic greedy algorithms based on a probabilistic interpretation of the squared norm of the a posteriori approximation error. This squared norm written as an expectation allows to select the next snapshot to enrich the current reduced space using the adaptive probably approximately correct algorithm in relative precision presented in Chapter 3. Our first algorithm used PAC algorithm for snapshots' selection on a fixed training set. Our second algorithm used PAC algorithm for snapshots' selection on random training sets [27]. Both algorithms offer a probabilistic selection of the snapshots and the second one is promising for high-dimensional parameter-dependent PDEs.

Further outlooks include goal-oriented reduced basis methods, such as it can be encountered in Uncertainty Quantification (UQ) when we look into some quantity of interest depending on the solution and not the solution itself. Using probabilistic interpretation of the solution to derive probabilistic goal-oriented reduced basis methods could be of high interest when the quantity of interest can be estimated using Monte-Carlo methods. It includes in particular quantities of interest such as integrals of the solution in a deterministic setting or statistical moments of some functional of the solution in a stochastic setting.

# Bibliography

- [1] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration–exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.
- [2] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Annual Conference on Learning Theory (COLT)*, 2010.
- [3] J.-Y. Audibert, S. Bubeck, and R. Munos. Bandit view on noisy optimization. *Optimization for Machine Learning*, page 431, 2011.
- [4] M. Bachmayr, R. Schneider, and A. Uschmajew. Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. *Foundations of Computational Mathematics*, 16(6):1423–1472, 2016.
- [5] V. Bally and D. Talay. The law of the euler scheme for stochastic differential equations. *Probability theory and related fields*, 104(1):43–60, 1996.
- [6] M. Barrault, Y. Maday, N. C. Nguyen, and A. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339(9):667–672, 2004.
- [7] C. Beck, S. Becker, P. Grohs, N. Jaafari, and A. Jentzen. Solving stochastic differential equations and kolmogorov equations by means of deep learning. *arXiv preprint arXiv:1806.00421*, 2018.
- [8] C. Beck, E. Weinan, and A. Jentzen. Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *Journal of Nonlinear Science*, 29(4):1563–1619, 2019.
- [9] A. Beskos and G. Roberts. Exact simulation of diffusions. *The Annals of Applied Probability*, 15(4):2422–2444, 2005.
- [10] H.-G. Beyer and B. Sendhoff. Robust optimization—a comprehensive survey. *Computer methods in applied mechanics and engineering*, 196(33-34):3190–3218, 2007.
- [11] M. Billaud-Friess, A. Macherey, A. Nouy, and C. Prieur. Stochastic methods for solving high-dimensional partial differential equations. In *International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, pages 125–141. Springer, 2018.

- [12] M. Billaud-Friess, A. Macherey, A. Nouy, and C. Prieur. A pac algorithm in relative precision for bandit problem with costly sampling. *arXiv preprint arXiv:2007.15331*, 2020.
- [13] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM journal on mathematical analysis*, 43(3):1457–1472, 2011.
- [14] M.-R. Blel, V. Ehrlacher, and T. Lelièvre. Influence of sampling on the convergence rates of greedy algorithms for parameter-dependent random variables. *arXiv preprint arXiv:2105.14091*, 2021.
- [15] M. Bossy, N. Champagnat, S. Maire, and D. Talay. Probabilistic interpretation and random walk on spheres algorithms for the poisson-boltzmann equation in molecular dynamics. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 44(5):997–1048, 2010.
- [16] S. Boyaval, T. Lelievre, et al. A variance reduction method for parametrized stochastic differential equations using the reduced basis paradigm. *Communications in Mathematical Sciences*, 8(3):735–762, 2010.
- [17] S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.
- [18] A. Buffa, Y. Maday, A. Patera, C. Prud’homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 46(3):595–603, 2012.
- [19] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta numerica*, 13(1):147–269, 2004.
- [20] Y. Chen, J. Jiang, and A. Narayan. A robust error estimator and a residual-free error indicator for reduced basis methods. *Computers & Mathematics with Applications*, 77(7):1963–1979, 2019.
- [21] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab. Sparse adaptive taylor approximation algorithms for parametric and stochastic elliptic pdes. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 47(1):253–280, 2013.
- [22] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric pdes. *Foundations of Computational Mathematics*, 14(4):601–633, 2014.
- [23] A. Chkifa, A. Cohen, and C. Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric pdes. *Journal de Mathématiques Pures et Appliquées*, 103(2):400–428, 2015.
- [24] A. Cohen and R. DeVore. Approximation of high-dimensional parametric pdes. *arXiv preprint arXiv:1502.06797*, 2015.

- [25] A. Cohen and R. DeVore. Kolmogorov widths under holomorphic mappings. *IMA Journal of Numerical Analysis*, 36(1):1–12, 2016.
- [26] A. Cohen, R. Devore, and C. Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic pde’s. *Analysis and Applications*, 9(01):11–47, 2011.
- [27] A. Cohen, W. Dahmen, R. DeVore, and J. Nichols. Reduced basis greedy selection using random training sets. *ESAIM: Mathematical Modelling and Numerical Analysis*, 54(5):1509–1524, 2020.
- [28] F. Comets and T. Meyre. Calcul stochastique et modeles de diffusions. *Dunod Paris*, 2006.
- [29] M. Deaconu and A. Lejay. A random walk on rectangles algorithm. *Methodology And Computing In Applied Probability*, 8(1):135, 2006.
- [30] M. Deaconu, S. Herrmann, and S. Maire. The walk on moving spheres: a new tool for simulating brownian motion’s exit time from a domain. *Mathematics and Computers in Simulation*, 135:28–38, 2017.
- [31] R. DeVore. Nonlinear approximation. *Acta numerica*, 7:51–150, 1998.
- [32] R. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in banach spaces. *Constructive Approximation*, 37(3):455–466, 2013.
- [33] V. Dupa and U. Herkenrath. Stochastic approximation on a discrete set and the multi-armed. *Sequential Analysis*, 1(1):1–25, 1982.
- [34] P. Étoré and M. Martinez. Exact simulation of one-dimensional stochastic differential equations involving the local time at zero of the unknown process. *Monte Carlo Methods and Applications*, 19(1):41–71, 2013.
- [35] P. Etoe and M. Martinez. Exact simulation for solutions of one-dimensional stochastic differential equations with discontinuous drift. *ESAIM: Probability and Statistics*, 18:686–702, 2014.
- [36] L. Evans. Graduate studies in mathematics. In *Partial differential equations*. Am. Math. Soc., 1998.
- [37] E. Even-Dar, S. Mannor, and Y. Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.
- [38] E. Even-Dar, S. Mannor, and Y. Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.
- [39] N. Flyer, G. Wright, and B. Fornberg. Radial basis function-generated finite differences: A mesh-free method for computational geosciences. *Handbook of geomathematics*, pages 1–30, 2014.
- [40] M. I. Freidlin. *Functional integration and partial differential equations*. Number 109. Princeton university press, 1985.



- [41] A. Friedman. Stochastic differential equations and applications. In *Stochastic Differential Equations*, pages 75–148. Springer, 2010.
- [42] N. Frikha and S. Menozzi. Concentration bounds for stochastic approximations. *Electronic Communications in Probability*, 17, 2012.
- [43] A. Garivier and O. Cappé. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *Proceedings of the 24th annual conference on learning theory*, pages 359–376, 2011.
- [44] M. Giles. Multilevel monte carlo methods. In *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 83–103. Springer, 2013.
- [45] J. Given, J. Hubbard, and J. Douglas. A first-passage algorithm for the hydrodynamic friction and diffusion-limited reaction rate of macromolecules. *The Journal of chemical physics*, 106(9):3761–3771, 1997.
- [46] E. Gobet. Introduction to stochastic calculus and to the resolution of pdes using monte carlo simulations. In *Advances in numerical simulation in physics and engineering*, pages 107–178. Springer, 2014.
- [47] E. Gobet. *Monte-Carlo methods and stochastic processes: from linear to non-linear*. CRC Press, 2016.
- [48] E. Gobet and S. Maire. A spectral monte carlo method for the poisson equation. *Monte Carlo methods and applications*, 10(3-4):275–285, 2004.
- [49] E. Gobet and S. Maire. Sequential control variates for functionals of markov processes. *SIAM Journal on Numerical Analysis*, 43(3):1256–1275, 2005.
- [50] E. Gobet and S. Menozzi. Stopped diffusion processes: boundary corrections and overshoot. *Stochastic Processes and Their Applications*, 120(2):130–162, 2010.
- [51] W.-B. Gong, Y.-C. Ho, and W. Zhai. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*, 10(2):384–404, 2000.
- [52] L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.
- [53] B. Haasdonk. Reduced basis methods for parametrized pdes—a tutorial introduction for stationary and instationary problems. *Model reduction and approximation: theory and algorithms*, 15:65, 2017.
- [54] B. Haasdonk and M. Ohlberger. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Mathematical and Computer Modelling of Dynamical Systems*, 17(2):145–161, 2011.
- [55] W. Hackbusch. Numerical tensor calculus. *Acta Numer.*, 23:651–742, 2014.
- [56] J. Han, A. Jentzen, and E. Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

- [57] S. Herrmann and E. Tanré. The first-passage time of the brownian motion to a curved boundary: an algorithmic approach. *SIAM Journal on Scientific Computing*, 38(1):A196–A215, 2016.
- [58] S. Herrmann and C. Zucca. Exact simulation of first exit times for one-dimensional diffusion processes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 54(3):811–844, 2020.
- [59] S. Herrmann and C. Zucca. Exact simulation of diffusion first exit times: algorithm acceleration. *arXiv preprint arXiv:2004.02313*, 2020.
- [60] J. Hesthaven, B. Stamm, and S. Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(1):259–283, 2014.
- [61] D. Higham, X. Mao, M. Roj, Q. Song, and G. Yin. Mean exit times and the multilevel monte carlo method. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):2–18, 2013.
- [62] C.-O. Hwang and M. Mascagni. Analysis and comparison of green’s function first-passage algorithms with “walk on spheres” algorithms. *Mathematics and computers in simulation*, 63(6):605–613, 2003.
- [63] C.-O. Hwang, J. Given, and M. Mascagni. The simulation–tabulation method for classical diffusion monte carlo. *Journal of Computational Physics*, 174(2):925–946, 2001.
- [64] B. Jourdain and A. Kebaier. Non-asymptotic error bounds for the multilevel monte carlo euler method applied to sdes with constant diffusion coefficient. *Electronic Journal of Probability*, 24, 2019.
- [65] M. Kac. On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):1–13, 1949.
- [66] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.
- [67] H. Kano, J. Honda, K. Sakamaki, K. Matsuura, A. Nakamura, and M. Sugiyama. Good arm identification via bandit feedback. *Machine Learning*, 108(5):721–745, 2019.
- [68] I. Karatzas and S. Shreve. Graduate texts in mathematics, 1991.
- [69] E. Kaufmann, O. Cappé, and A. Garivier. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1):1–42, 2016.
- [70] B. Khoromskij. Tensor numerical methods for multidimensional pdes: theoretical analysis and initial applications. *ESAIM: Proceedings and Surveys*, 48:1–28, 2015.
- [71] P. Kloeden and E. Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 2013.

- [72] R. Kunsch. High-dimensional function approximation: Breaking the curse with monte carlo methods. *arXiv preprint arXiv:1704.08213*, 2017.
- [73] T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.
- [74] A. Locatelli, M. Gutzeit, and A. Carpentier. An optimal algorithm for the thresholding bandit problem. *arXiv preprint arXiv:1605.08671*, 2016.
- [75] Y. Maday, N. C. Nguyen, A. Patera, and S. H. Pau. A general multipurpose interpolation procedure: the magic points. *Communications on Pure & Applied Analysis*, 8(1):383, 2009.
- [76] M. Mascagni and C.-O. Hwang.  $\epsilon$ -shell error analysis for “walk on spheres” algorithms. *Mathematics and computers in simulation*, 63(2):93–104, 2003.
- [77] J. Michael. Barriers for uniformly elliptic equations and the exterior cone condition. *Journal of Mathematical Analysis and Applications*, 79(1):203–217, 1981.
- [78] V. Mnih. *Efficient stopping rules*. PhD thesis, University of Alberta, 2008.
- [79] V. Mnih, C. Szepesvári, and J.-Y. Audibert. Empirical bernstein stopping. In *Proceedings of the 25th international conference on Machine learning*, pages 672–679, 2008.
- [80] M. Motoo. Some evaluations for continuous monte carlo method by using brownian hitting process. *Annals of the Institute of Statistical Mathematics*, 11(1):49–54, 1959.
- [81] S. Mukherjee, K. P. Naveen, N. Sudarsanam, and B. Ravindran. Thresholding bandits with augmented ucb. *arXiv preprint arXiv:1704.02281*, 2017.
- [82] M. Muller. Some continuous monte carlo methods for the dirichlet problem. *The Annals of Mathematical Statistics*, 27(3):569–589, 1956.
- [83] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.
- [84] A. Nouy. Low-rank methods for high-dimensional approximation and model order reduction. *Model reduction and approximation*, P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, eds., SIAM, Philadelphia, PA, pages 171–226, 2017.
- [85] E. Novak and H. Woźniakowski. *Tractability of Multivariate Problems: Standard information for functionals*, volume 2. European Mathematical Society, 2008.
- [86] E. Novak and H. Woźniakowski. Approximation of infinitely differentiable multivariate functions is intractable. *Journal of Complexity*, 25(4):398–404, 2009.
- [87] I. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

- [88] I. Oseledets and E. Tyrtysnikov. Breaking the curse of dimensionality, or how to use svd in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, 2009.
- [89] G. Pagès. Multi-step richardson-romberg extrapolation: remarks on variance control and complexity. *Monte Carlo Methods and Applications*, 13(1):37–70, 2007.
- [90] A. Pinkus. *N-widths in Approximation Theory*, volume 7. Springer Science & Business Media, 2012.
- [91] H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.
- [92] G. Rozza, D. B. P. Huynh, and A. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1, 2007.
- [93] F. Salmoiraghi, F. Ballarin, G. Corsi, A. Mola, M. Tezzele, and G. Rozza. Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: overview and perspectives. In *VII European Congress on Computational Methods in Applied Sciences and Engineering*, volume 1, pages 1013–1031. Institute of Structural Analysis and Antiseismic Research School of Civil Engineering, 2016.
- [94] J. Shen and H. Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic problems. *SIAM Journal on Scientific Computing*, 32(6):3228–3250, 2010.
- [95] J. Shen and H. Yu. Efficient spectral sparse grid methods and applications to high-dimensional elliptic equations ii. unbounded domains. *SIAM Journal on Scientific Computing*, 34(2):A1141–A1164, 2012.
- [96] J. Strikwerda. *Finite difference schemes and partial differential equations*. SIAM, 2004.
- [97] B. Sudret, S. Marelli, and J. Wiart. Surrogate models for uncertainty quantification: An overview. In *2017 11th European conference on antennas and propagation (EUCAP)*, pages 793–797. IEEE, 2017.
- [98] R. Sutton and A. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [99] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic analysis and applications*, 8(4):483–509, 1990.
- [100] C. Tao, S. Blanco, J. Peng, and Y. Zhou. Thresholding bandit with optimal aggregate regret. In *Advances in Neural Information Processing Systems*, pages 11664–11673, 2019.
- [101] E. Weinan, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward

- stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.
- [102] D. Yan and H. Mukai. Stochastic discrete optimization. *SIAM Journal on control and optimization*, 30(3):594–612, 1992.
- [103] O. Zahm and A. Nouy. Interpolation of inverse operators for preconditioning parameter-dependent equations. *SIAM Journal on Scientific Computing*, 38(2):A1044–A1074, 2016.
- [104] O. C. Zienkiewicz and R. L. Taylor. *The finite element method, vol. 2*. Butterworth-Heinemann, 2000.



---

**Titre :** Approximation et réduction de modèle pour les équations aux dérivées partielles avec interprétation probabiliste

**Mot clés :** équations aux dérivées partielles, approximation en grande dimension, méthodes de bases réduites, stratégies adaptatives, méthodes de Monte-Carlo, optimisation discrète

**Résumé :** Nous nous intéressons dans cette thèse à la résolution numérique de modèles régis par des équations aux dérivées partielles admettant une interprétation probabiliste.

Dans un premier temps, nous considérons des équations aux dérivées partielles en grande dimension. En nous basant sur une interprétation probabiliste de la solution qui permet d'obtenir des évaluations ponctuelles de celle-ci via des méthodes de Monte-Carlo, nous proposons un algorithme combinant une méthode d'interpolation adaptative et une méthode de réduction de variance pour approcher la solution sur tout son domaine de définition.

Dans un deuxième temps, nous nous intéressons aux méthodes de bases réduites pour les équations aux dérivées partielles paramétrées. Nous proposons deux algorithmes gloutons reposant sur une interprétation probabiliste de l'erreur. Nous proposons également un algorithme d'optimisation discrète *probably approximately correct* en précision relative qui nous permet, pour ces deux algorithmes gloutons, de sélectionner judicieusement un *snapshot* à ajouter à la base réduite en se basant sur la représentation probabiliste de l'erreur d'approximation.

---

**Title:** Approximation and model reduction for partial differential equations with probabilistic interpretation

**Keywords:** partial differential equations, high-dimensional approximation, reduced basis methods, adaptive strategies, Monte-Carlo methods, discrete optimization

**Abstract:** In this thesis, we are interested in the numerical solution of models governed by partial differential equations that admit a probabilistic interpretation.

In a first part, we consider partial differential equations in high dimension. Based on a probabilistic interpretation of the solution which allows to obtain pointwise evaluations of the solution using Monte-Carlo methods, we propose an algorithm combining an adaptive interpolation method and a variance reduction method to approximate the global solution.

In a second part, we focus on reduced basis methods for parametric partial differential equations. We propose two greedy algorithms based on a probabilistic interpretation of the error. We also propose a discrete optimization algorithm *probably approximately correct* in relative precision which allows us, for these two greedy algorithms, to judiciously select a *snapshot* to add to the reduced basis based on the probabilistic representation of the approximation error.