

Algorithmes de simulation de champs gaussiens

Valentin Abongo

27 septembre 2023

Résumé

On s'intéresse ici à la simulation de champs gaussiens. Il existe de multiples façons de les simuler et l'objectif sera de faire un inventaire sur plusieurs méthodes. Les simulations qui y sont décrites ont été testées en code python à l'aide d'une bibliothèque C++/Python du nom d'OpenTURNS¹ dont Airbus est l'un des contributeurs. OpenTURNS permet notamment le traitement des incertitudes à l'aide de diverses notions de statistique et de modéliser des phénomènes aléatoires, dont les champs gaussiens. Par conséquent certaines méthodes de simulation y sont déjà implémentées et ne nécessitent donc que de les trouver dans la bibliothèque. Cependant d'autres méthodes n'existaient pas en soi sur OpenTURNS et elles ont mené à produire du nouveau code basé essentiellement sur la bibliothèque.

1. <http://openturns.github.io/>

Table des matières

Table des matières	2
1 Introduction	5
2 Simulation par méthode de Cholesky	7
2.1 Méthode de Cholesky	7
2.2 Estimation de l'erreur	9
2.2.1 Quantification de l'erreur	9
2.2.2 Exemple sur la méthode de Cholesky	10
2.3 Erreur et convergence numérique	11
2.3.1 Dimension $n=1$	11
2.3.2 Dimension $n=2$	11
2.3.3 Dimension $n=3$	12
2.3.4 Benchmarks	13
3 Optimisation par les \mathcal{H}-matrices	15
3.1 Idée globale des \mathcal{H} -matrices	15
3.2 Compression en \mathcal{H} -matrice	16
3.2.1 Découpage hiérarchique	16
3.2.1.1 Découpage hiérarchique de l'espace	16
3.2.1.2 Découpage hiérarchique de la matrice	17
3.2.2 Compression des blocs admissibles	19
3.3 Complexité spatiale et temporelle	21
3.4 Erreur et convergence numérique	22
3.4.1 Paramétrage pour les tests	22
3.4.2 Benchmarks	22
4 Méthode de Galli-Gao-Gibbs	25
4.1 Échantillonnage de Gibbs	25
4.2 Cas gaussien	27
4.2.1 Méthode de Gibbs	27
4.2.2 Méthode de Galli-Gao-Gibbs	28
4.3 Complexité algorithmique	31
4.4 Erreur et convergence numérique	31

4.4.1	Dimension $n=1$	31
4.4.2	Benckmarks	31
5	Simulation par méthode spectrale	33
5.1	Hypothèses sur la densité spectrale	33
5.2	Factorisation de Cholesky	34
5.3	Subdivision du domaine spectral	34
5.4	Subdivision du domaine de simulation	36
5.5	Approximation du champ gaussien X	37
5.6	Algorithme de simulation	40
5.6.1	Premier algorithme	40
5.6.2	Reformulation avec la densité spectrale fréquentielle . .	40
5.6.3	Autre reformulation de l'approximation	41
5.7	Algorithme de simulation et TFD	42
5.8	Propriétés des approximations X_N	43
5.9	Erreur et convergence numérique	45
5.9.1	Nouvel estimateur	45
5.9.2	Résultats numériques	46
5.9.2.1	Dimension $n=1$	47
5.9.2.2	Dimension $n=2$	47
5.9.2.3	Dimension $n=3$	48
5.9.2.4	Benchmarks	48
6	Simulation par \mathbb{P}_1-interpolation	51
6.1	Cadre théorique	51
6.2	Approximation d'une fonction par méthode des éléments finis \mathbb{P}_1	52
6.3	Espaces de Sobolev et interpolation	53
6.4	Espaces de Sobolev et champs aléatoires	54
6.4.1	Sur la différentiabilité en moyenne quadratique d'un champ aléatoire	54
6.4.2	Résultat de régularité des trajectoires	56
6.5	Méthode de \mathbb{P}_1 -interpolation	57
6.5.1	Description de la méthode	57
6.5.2	Estimation d'erreur	58
6.6	Erreur et convergence numérique	60
6.6.1	Dimension $n=2$	60
6.6.2	Dimension $n=3$	62
6.6.3	Benchmarks	63
	Bibliographie	67
A	Base théorique pour la méthode spectrale	69
A.1	Mesure matricielle	69
A.2	Mesure stochastique orthogonale centrée et isométrie	70

A.3	Sur l'existence d'une m.s.o.c associée à un champ aléatoire . . .	71
A.3.1	Mesure spectrale	72
A.3.2	Densité spectrale	72
A.3.3	Densité spectrale fréquentielle	73
A.4	Représentation spectrale de X	73
B	Code pour les tests numériques	75
B.1	Dossier CholeskyMethod	75
B.2	Dossier HmatrixMethod	75
B.3	Dossier GalliGaoGibbsMethod	75
B.4	Dossier SpectralMethod	76
B.5	Dossier P1InterpolationMethod	78

Chapitre 1

Introduction

Problème général

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé

Soit n, d deux entiers naturels non nuls et D un fermé borné de \mathbb{R}^n

Soit $X = (X_t)_{t \in \mathbb{R}^n}$ un processus gaussien centré d'ordre 2 à valeurs dans \mathbb{R}^d

Notons $C : \mathbb{R}^n \times \mathbb{R}^n \rightarrow M_d(\mathbb{R})$ la fonction de covariance de X

On souhaite simuler une réalisation du processus X sur le domaine D . Pour cela, on discrétise le domaine D par un maillage M (de préférence une triangulation de D) et on simule X sur les nœuds du maillage $(n_i)_{i \in \llbracket 1; m \rrbracket}$ où m est le nombre de nœuds. Il s'agit donc de simuler le vecteur aléatoire $X_M = (X_{n_1}, \dots, X_{n_m})$, qui n'est rien d'autre qu'un vecteur gaussien centré à valeurs dans $(\mathbb{R}^d)^m$ qu'on assimile à \mathbb{R}^{dm} . D'où X_M est caractérisé par sa matrice de covariance $\Sigma \in M_{dm}(\mathbb{R})$ dont les coefficients sont donnés par la fonction de covariance C . Plus concrètement pour $(i, k) \in \llbracket 1; m \rrbracket^2, (j, l) \in \llbracket 1; d \rrbracket^2$, $\Sigma_{(i-1)m+j, (k-1)m+l} = C(n_i, n_k)_{j,l} = \mathbb{E}((X_{n_i})_j (X_{n_k})_l)$. On a en particulier que Σ est symétrique semi-définie positive.

La problématique revient donc à simuler la loi $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$. On présentera d'abord des méthodes classiques pour ensuite en exhiber d'autres dont l'approche considère davantage le type de processus que l'on cherche à simuler.

On supposera par la suite que Σ est symétrique réelle définie positive, quitte à la régulariser¹.

1. En notant λ_{max} le module maximum du spectre de Σ , la régularisation se fait en ajoutant une fraction ϵ de λ_{max} sur sa diagonale i.e on ajoute $\epsilon \cdot \lambda_{max}$ aux coefficients diagonaux de Σ

Chapitre 2

Simulation par méthode de Cholesky

2.1 Méthode de Cholesky

En utilisant les notations du chapitre 1, évoquons deux propriétés qui conduisent au premier algorithme de simulation de la gaussienne $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$.

Propriété 2.1.1. Soit $r \in \mathbb{N}^*$ et U_1, \dots, U_r r variables aléatoires réelles iid suivant la loi gaussienne centrée réduite $\mathcal{N}(0, 1)$. Alors :

$$U = (U_1, \dots, U_r) \sim \mathcal{N}(0_{\mathbb{R}^r}, I_r)$$

Propriété 2.1.2. Soit $r \in \mathbb{N}^*$, $V : \Omega \rightarrow \mathbb{R}^r$ un vecteur aléatoire tel que $V \sim \mathcal{N}(0_{\mathbb{R}^r}, I_r)$, A une matrice de $M_r(\mathbb{R})$, et L une matrice triangulaire inférieure de $M_r(\mathbb{R})$ tel que

$$A = LL^T \tag{2.1}$$

$$\text{Alors : } LV \sim \mathcal{N}(0_{\mathbb{R}^r}, A)$$

Remarque 2.1.1. La relation (2.1) est la décomposition de Cholesky de A par L . De ces propriétés, on obtient que si Σ admet une décomposition de Cholesky et que l'on sait simuler des réalisations indépendantes de la loi gaussienne univariée centrée réduite, alors on peut simuler $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$.

Par une méthode du style Box-Muller, on sait simuler des réalisations indépendantes de la loi gaussienne univariée centrée réduite. Puis comme Σ est symétrique définie positive, Σ admet une décomposition de Cholesky LL^T . Pour obtenir cette décomposition, on utilise la méthode de Cholesky qu'on peut appliquer à toute matrice réelle symétrique définie-positive. (voir **Algorithme 1**). La décomposition de Cholesky est unique du moment que l'on impose à ce que les coefficients diagonaux de L soient des réels positifs, ce que garantit la méthode de Cholesky.

Finalement si on note $\Sigma = LL^T$, la décomposition de Cholesky de Σ , on déduit un algorithme de simulation de N réalisations indépendantes de X_M (voir **Algorithme 2**).

Algorithme 1 MÉTHODE DE CHOLESKY

Entrée(s) A une matrice symétrique définie positive de taille r

Début

Soit L une matrice nulle de taille r

pour $j := 1$ **à** r **faire**

$$L[j, j] = \sqrt{A[j, j] - \sum_{k=1}^{j-1} |L[j, k]|^2}$$

pour $i := j+1$ **à** r **faire**

$$L[i, j] = \frac{A[i, j] - \sum_{k=1}^{j-1} L[j, k]L[i, k]}{L[j, j]}$$

fin du pour

fin du pour

Fin

Sortie(s) L

Algorithme 2 ALGORITHME DE SIMULATION

Entrée(s) x liste de longueur N (nombre de réalisations), $L \in M_{dm}(\mathbb{R})$

Début

pour $i := 1$ **à** N **faire**

on produit $y = (y_1, \dots, y_{dm})^T \in \mathbb{R}^{dm}$ selon $\mathcal{N}(0_{\mathbb{R}^{dm}}, I_{dm})$

$x[i] = Ly \in \mathbb{R}^{dm}$ (produit matriciel de L et y)

fin du pour

Fin

Sortie(s) x

En observant l'algorithme 2, on saisit que la complexité en temps d'une réalisation du vecteur gaussien $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$ dépend de si la matrice L a été calculée ou non. Si c'est le cas, cette complexité se veut en $O((dm)^2)$ (produit matrice-vecteur). Si ce n'est pas le cas, il faut aussi calculer la factorisation de Cholesky et la complexité en temps devient $O((dm)^3)$. La complexité mémoire est en $O((dm)^2)$ (stockage de Σ).

2.2 Estimation de l'erreur

2.2.1 Quantification de l'erreur

Par la suite pour $r \in \mathbb{N}^*$, \mathbb{R}^r pourra être vu comme l'ensemble des matrices colonnes à r lignes.

Pour savoir si le vecteur aléatoire X_M est bien simulé, il faut quantifier une erreur de simulation. La loi forte des grands nombres stipule que si l'on considère r un entier non nul, $(Y_N)_{N \in \mathbb{N}^*}$ une suite de vecteurs aléatoires dans $L^1_{\mathbb{R}^r}(\Omega, \mathbb{P})$ iid selon la loi d'un élément $Y \in L^1_{\mathbb{R}^r}(\Omega, \mathbb{P})$, alors :

$$\frac{1}{N} \sum_{i=1}^N Y_i - \mathbb{E}(Y) \xrightarrow[N \rightarrow \infty]{p.s} 0$$

Par conséquent si on considère $(Y_N)_{N \in \mathbb{N}^*}$ la suite de vecteurs aléatoires modélisant des réalisations indépendantes du vecteur aléatoire X_M , on a :

$$\frac{1}{N} \sum_{i=1}^N Y_i Y_i^T \xrightarrow[N \rightarrow \infty]{p.s} \mathbb{E}(X_M X_M^T) = \Sigma$$

Donc si on munit $M_{dm}(\mathbb{R})$ de la norme de Frobenius $\|\cdot\|_F$, on obtient la reformulation suivante :

$$\left\| \frac{1}{N} \sum_{i=1}^N Y_i Y_i^T - \Sigma \right\|_F \xrightarrow[N \rightarrow \infty]{p.s} 0$$

De ce résultat, on va pouvoir quantifier l'erreur de N réalisations du vecteur $X_M \sim \mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$, $y = (y_1, \dots, y_N) \in (\mathbb{R}^{dm})^N$, en introduisant la définition suivante.

Définition 2.2.1. Soit N un entier naturel non nul et $(z_1, \dots, z_N) \in (\mathbb{R}^{dm})^N$. On définit alors l'erreur L^2 (relative à X_M) des N réalisations $z = (z_1, \dots, z_N)$ par la quantité :

$$\epsilon(\Sigma, z) = \frac{\left\| \frac{1}{N} \sum_{i=1}^N z_i z_i^T - \Sigma \right\|_F}{\|\Sigma\|_F}$$

Ainsi, si on simule bien X_M , on a forcément par la loi forte des grands nombres, que plus le nombre de réalisations N est grand, plus $\epsilon(\Sigma, y)$ doit tendre vers 0. L'erreur L^2 est donc un moyen de vérifier numériquement si notre façon de modéliser le vecteur X_M est correct.

2.2.2 Exemple sur la méthode de Cholesky

Présentons un exemple concret où l'erreur L^2 a été calculée. On simule via Cholesky un processus gaussien centré d'ordre 2 où $n = d = 1$ et ayant pour fonction de covariance $C(x, y) = e^{-|x-y|}$ pour x, y dans \mathbb{R} . On choisit ici $D = [-10, 10]$ et comme maillage M , une subdivision régulière de D en 1000 intervalles (M possède donc 1001 nœuds). Dans la figure 2.1, une réalisation de X_M a été faite en les points du maillage.

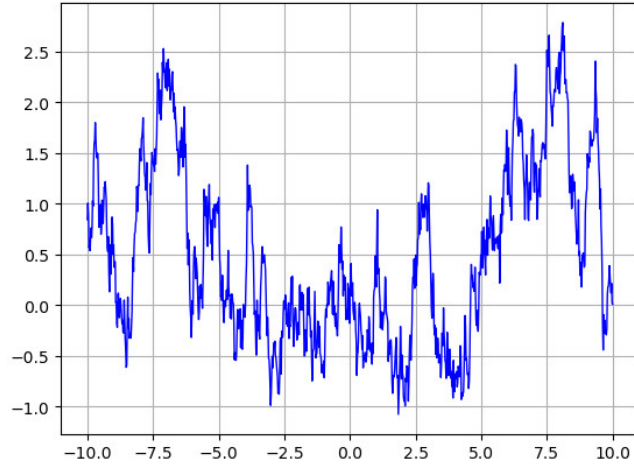


FIGURE 2.1 – Réalisation du processus gaussien sur $[-10, 10]$

Nb de réalisations	erreur L^2
125	0.45627
250	0.27285
500	0.19814
750	0.16260
1000	0.15045

Erreur L^2 en fonction du nombre de réalisations

Remarque 2.2.1. Le trait de la courbe de la figure 2.1 se veut continue par interpolation linéaire en les points du maillage qui ont une valeur.

2.3 Erreur et convergence numérique

La méthode de Cholesky nécessite un maillage M et une fonction de covariance C pour déduire la matrice de covariance Σ . On se propose ici de décrire les maillages et les fonctions de covariance utilisées pour les tests numériques.

2.3.1 Dimension $n=1$

Pour les simulations, on a choisi $D = [-10, 10]$. Pour $m \in \mathbb{N}^*$, $m > 1$, les maillages considérés seront caractérisés à l'aide des nœuds du type

$$\left(t_i = -10 + \frac{20i}{m-1} \right)_{i \in \llbracket 0; m-1 \rrbracket}$$

et les éléments d'un maillage seront les intervalles de la forme $[t_i, t_{i+1}]$, $i \in \llbracket 0; m-2 \rrbracket$. On fera varier le nombre de nœuds m entre 10 et 10000. On n'associera à ces maillages qu'une seule fonction de covariance C :

$$C(x, y) = \exp(-|y - x|) \text{ pour } (x, y) \in \mathbb{R}^2$$

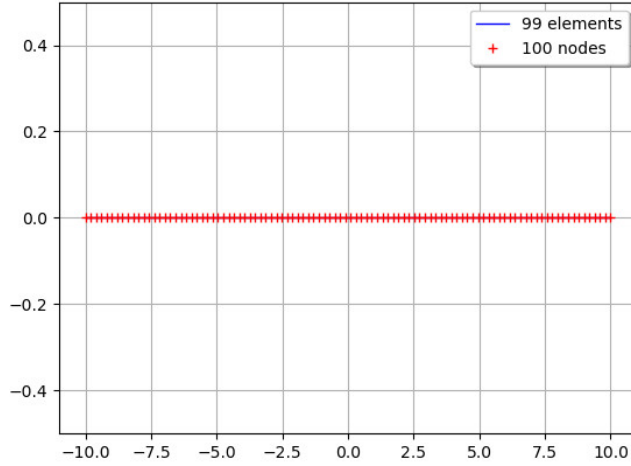


FIGURE 2.2 – Maillage à 100 nœuds en dimension 1

2.3.2 Dimension $n=2$

En dimension 2, on a choisi $D = [-10, 10]^2$. Pour $m \in \mathbb{N}^*$, $m > 1$, les maillages considérés seront caractérisés par des nœuds du type

$$\left(t_{i,j} = \left(-10 + \frac{20i}{m-1}, -10 + \frac{20j}{m-1} \right) \right)_{(i,j) \in \llbracket 0; m-1 \rrbracket^2}$$

et par leurs éléments triangulaires dont les sommets sont décrits par des triplets du type

$$(t_{i,j}, t_{i+1,j}, t_{i,j+1}) \text{ ou } (t_{i+1,j}, t_{i+1,j+1}, t_{i,j+1}), (i, j) \in \llbracket 0; m-2 \rrbracket^2$$

On fera varier le nombre de nœuds m^2 entre 10 et 10000.

On n'associera à ces maillages qu'une seule fonction de covariance C :

$$C(x, y) = \exp(-\|y - x\|_2) \text{ pour } (x, y) \in (\mathbb{R}^2)^2$$

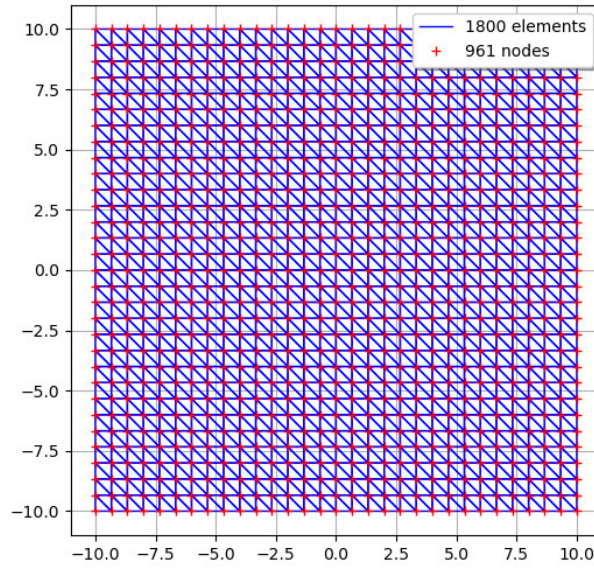


FIGURE 2.3 – Maillage à 961 nœuds en dimension 2

2.3.3 Dimension n=3

En dimension 3, on a choisi $D = [-10, 10]^3$. Pour $m \in \mathbb{N}^*$, $m > 1$, les maillages considérés seront caractérisés par des nœuds du type

$$\left(t_{i,j,k} = \left(-10 + \frac{20i}{m-1}, -10 + \frac{20j}{m-1}, -10 + \frac{20k}{m-1} \right) \right)_{(i,j,k) \in \llbracket 0; m-1 \rrbracket^3}$$

et par leurs éléments tétraédriques dont les sommets sont décrits pour $(i, j, k) \in \llbracket 0; m-2 \rrbracket^3$ par des quadruplets ayant une de ces 6 formes :

$$(t_{i,j,k}, t_{i+1,j,k}, t_{i+1,j,k+1}, t_{i+1,j+1,k+1}),$$

$$(t_{i,j,k}, t_{i+1,j+1,k}, t_{i+1,j,k}, t_{i+1,j+1,k+1}),$$

$$\begin{aligned} &(t_{i,j,k}, t_{i+1,j,k+1}, t_{i,j,k+1}, t_{i+1,j+1,k+1}), \\ &(t_{i,j,k}, t_{i,j,k+1}, t_{i,j+1,k+1}, t_{i+1,j+1,k+1}), \\ &(t_{i,j,k}, t_{i,j+1,k+1}, t_{i,j+1,k}, t_{i+1,j+1,k+1}), \\ &(t_{i,j,k}, t_{i,j+1,k}, t_{i+1,j+1,k}, t_{i+1,j+1,k+1}) \end{aligned}$$

On fera varier le nombre de nœuds m^3 entre 10 et 10000.

On n'associera à ces maillages qu'une seule fonction de covariance C :

$$C(x, y) = \exp(-\|y - x\|_2) \text{ pour } (x, y) \in (\mathbb{R}^3)^2$$

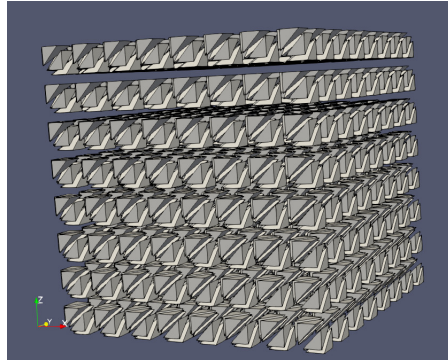


FIGURE 2.4 – Maillage à 729 nœuds en dimension 3

2.3.4 Benchmarks

Dimension	Nb de nœuds	Temps moyen d'une réalisation (en secondes)
1	10	0.00020s
1	100	0.00054s
1	1000	0.16301s
1	10000	183.51s
2	9	0.00075s
2	100	0.00084s
2	961	0.19619s
2	10000	164.28s
3	8	0.00044s
3	64	0.00060s
3	729	0.05580s
3	9261	116.86s

Remarque 2.3.1. Le nombre moyen de réalisations est estimé sur une durée de 5 secondes maximum. Une durée supérieure à 5 secondes pointe le fait qu'une seule réalisation a pu être effectuée et qu'elle a duré plus de 5 secondes. Certes pour un grand maillage, la durée non négligeable de la factorisation de Cholesky de Σ augmente le temps de la première réalisation, mais bien souvent les capacités limitées de la mémoire vive nécessitent de la mémoire supplémentaire et l'accès à cette mémoire ralentit l'exécution du code. Ceci explique notamment les temps moyens supérieures à 100 secondes. On note aussi que pour un maillage donnée, l'erreur L^2 tend vers 0 comme l'inverse de la racine carrée du nombre de réalisations.

Dimension	Nb de nœuds	Nb de réalisations	erreur L^2
1	10	250	0.14875
1	10	500	0.12409
1	10	750	0.09597
1	10	1000	0.07025
1	100	250	0.26887
1	100	500	0.19460
1	100	750	0.14790
1	100	1000	0.13010
1	1000	250	0.29682
1	1000	500	0.18924
1	1000	750	0.17071
1	1000	1000	0.14616
2	9	250	0.20574
2	9	500	0.12867
2	9	750	0.09212
2	9	1000	0.09902
2	100	250	0.45563
2	100	500	0.31861
2	100	750	0.26100
2	100	1000	0.22472
2	961	250	0.91385
2	961	500	0.65045
2	961	750	0.53042
2	961	1000	0.45776
3	8	250	0.15579
3	8	500	0.11870
3	8	750	0.09126
3	8	1000	0.08644
3	64	250	0.37056
3	64	500	0.25594
3	64	750	0.21249
3	64	1000	0.18017
3	729	250	1.1991
3	729	500	0.84544
3	729	750	0.69187
3	729	1000	0.59801

Chapitre 3

Optimisation par les \mathcal{H} -matrices

Si on reconsidère l'algorithme 2 de la section 2.1, et que l'on pose $r = dm$, on rappelle la nécessité de connaître la factorisation de Cholesky L de la matrice de covariance $\Sigma \in M_r(\mathbb{R})$ pour simuler $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$. Contrairement aux autres opérations qui permettent la simulation, la méthode de Cholesky qui déduit L à partir de Σ est très coûteuse : le nombre d'opérations est en $\mathcal{O}(r^3)$. Ce temps nécessaire pour déterminer L sera appelé le coût de démarrage. On devine clairement que pour un r suffisamment grand (de l'ordre de 10^3), le coût de démarrage devient non négligeable. Par conséquent, si on veut simuler une gaussienne, il peut vite s'avérer judicieux de réduire ce coût. De multiples façons existent pour le faire. On se propose ici de présenter l'une d'elles qui consiste à considérer une structure de données qui peut permettre dans certains cas la réduction du coût de démarrage. Il s'agit de la structure des matrices hiérarchiques ou encore \mathcal{H} -matrices. Elles permettent des représentations compressées de matrices denses. Cette structure ne sera pas décrite dans les détails mais tout lecteur intéressé pourra se référer à la thèse de Benoît Lizé [9].

3.1 Idée globale des \mathcal{H} -matrices

L'idée de base de la structure des \mathcal{H} -matrices est de pouvoir approcher certains blocs d'une matrice par des matrices de la forme AB^T . Une représentation de la forme AB^T peut réduire le coût mémoire de la matrice. En effet si on considère $A \in M_{m,k}(\mathbb{R})$ et $B \in M_{n,k}(\mathbb{R})$ où $k \leq \min(m, n)$, alors stocker AB^T coûte en mémoire mn cases-mémoire alors que stocker A et B coûte $k(m + n)$, ce qui peut être intéressant si $k \ll \min(m, n)$.

La représentation compressée d'une matrice dense C en \mathcal{H} -matrice nécessite deux étapes : une subdivision en plusieurs blocs de la matrice C (découpage hiérarchique) puis on compresse au format AB^T les blocs qui sont admissibles.

Une fois que l'on obtient la représentation compressée de la matrice, il sera possible de faire diverses opérations élémentaires d'algèbre linéaire tout en préservant la structure de \mathcal{H} -matrice. Dans le cas de la simulation d'un champ

gaussien, vu que l'algorithme de simulation est le même que l'algorithme 2 de la section 2.1, sauf que l'on utilise la structure des \mathcal{H} -matrices, deux opérations seront différentes dans leur implémentation (qu'on n'évoquera pas) : la factorisation de Cholesky et le produit matrice-vecteur.

Pour la suite, pour $M \in M_{p,q}(\mathbb{R})$, $\sigma \subset \llbracket 1;p \rrbracket$, $\tau \subset \llbracket 1;q \rrbracket$, on note $M_{|\sigma \times \tau}$ le bloc $\sigma \times \tau$ de la matrice M , $(M_{i,j})_{(i,j) \in \sigma \times \tau}$. On dira aussi que σ est un ensemble d'indices « ligne » et que τ est un ensemble d'indices « colonne ».



FIGURE 3.1 – Compression au format AB^T

3.2 Compression en \mathcal{H} -matrice

3.2.1 Découpage hiérarchique

Dans cette sous-section, on emploiera du vocabulaire propre à la théorie des graphes, notamment le concept d'arbre. À la page 66 de la thèse de Lizé, il est possible de se familiariser avec ce vocabulaire.

Considérons m, d et N_{leaf} 3 entiers non nuls, $M \in M_m(\mathbb{R})$ une matrice carrée. On suppose de plus qu'à chaque indice j de l'ensemble $\llbracket 1;m \rrbracket$, on associe un vecteur $x^j \in \mathbb{R}^d$ (dont on note sa $i^{\text{ème}}$ coordonnée $(x^j)_i$). On peut voir les x^j comme associés aux nœuds d'un maillage mais on ne va pas jusqu'à supposer que les x^i sont deux à deux distincts. Notons par ailleurs $\mathcal{I} = \llbracket 1;m \rrbracket$.

Remarque 3.2.1. Si on réutilise les notations du chapitre 1 pour le temps de cette remarque et que l'on reconsidère la matrice de covariance $\Sigma \in M_{dm}(\mathbb{R})$, en raison de sa définition, pour chaque indice i de l'ensemble $\llbracket 1;dm \rrbracket$, on peut associer de façon logique un nœud du maillage M . En effet pour $i \in \llbracket 1;m \rrbracket$, $j \in \llbracket 1;d \rrbracket$, on associe à l'indice $(i-1)m + j \in \llbracket 1;dm \rrbracket$, le nœud n_i .

3.2.1.1 Découpage hiérarchique de l'espace

Pour cette première étape, l'idée est de construire récursivement à partir de la racine, un arbre $\mathcal{T}_{\mathcal{I}}$ dont les sommets sont des parties de $\llbracket 1;m \rrbracket$ (donc des ensembles d'indices). L'arbre obtenu est un Cluster Tree (ou arbre de groupes). La racine de l'arbre sera l'ensemble $\llbracket 1;m \rrbracket$. Afin de construire l'arbre, on a besoin d'une fonction « Split » qui à partir d'un ensemble d'indices σ , produit une partition de σ , $\sigma_1, \dots, \sigma_l$, où l est en général fixé. Quand l vaut 2 on

obtient un arbre binaire. Par ailleurs, on contraint à ce que toutes les feuilles de $\mathcal{T}_{\mathcal{I}}$ soient des ensembles d'indices de cardinal inférieur ou égal à N_{leaf} .

La construction du Cluster Tree est donnée par l'algorithme 3 où pour un sommet σ de l'arbre, on notera $S(\sigma)$ l'ensemble des fils du sommet σ . Donc en particulier, si σ est une feuille, $S(\sigma)$ est l'ensemble vide.

Remarque 3.2.2. Les feuilles de $\mathcal{T}_{\mathcal{I}}$ forment une partition de \mathcal{I} .

Algorithme 3 CRÉATION DU CLUSTER TREE

```

déf createClusterTree( $\sigma$ )
  si  $|\sigma| \leq N_{leaf}$  alors
     $S(\sigma) = \emptyset$ 
  sinon
     $\sigma_1, \dots, \sigma_l = \text{Split}(\sigma)$ 
     $S(\sigma) = \{\sigma_1, \dots, \sigma_l\}$ 
    createClusterTree( $\sigma_1$ )
    ...
    createClusterTree( $\sigma_l$ )

```

Fin

Pour la fonction « Split » qui forme une partition, on évoquera deux choix possibles de fonction : le découpage médian et le découpage géométrique. Si on considère $\sigma \subset \llbracket 1; m \rrbracket$, le début des deux découpages est similaire : posons pour $i \in \llbracket 1; d \rrbracket$, $x_{min, \sigma}^i = \min_{j \in \sigma} (x^j)_i$ et $x_{max, \sigma}^i = \max_{j \in \sigma} (x^j)_i$. La première étape consiste à déterminer $i^* \in \llbracket 1; d \rrbracket$ tel que $x_{max, \sigma}^{i^*} - x_{min, \sigma}^{i^*}$ soit de valeur maximale (on détermine donc l'axe où la boîte englobante associée au $(x^j)_{j \in \sigma}$ est de largeur maximale).

La seconde étape produit la partition de σ . Pour le découpage médian, on obtient la partition de σ , σ_1 et σ_2 , telle que si on note $x_{med, \sigma}^{i^*}$ la médiane de l'ensemble $\{(x^j)_{i^*}, j \in \sigma\}$, alors pour $k \in \sigma$, k est dans σ_1 si $(x^k)_{i^*} \leq x_{med, \sigma}^{i^*}$, sinon k est dans σ_2 . Pour le découpage géométrique, on obtient la partition de σ , σ_1 et σ_2 , telle que si on note $x_{geo, \sigma}^{i^*} = (x_{min, \sigma}^{i^*} + x_{max, \sigma}^{i^*})/2$ alors pour $k \in \sigma$, k est dans σ_1 si $(x^k)_{i^*} \leq x_{geo, \sigma}^{i^*}$, sinon k est dans σ_2 .

3.2.1.2 Découpage hiérarchique de la matrice

Le découpage hiérarchique de la matrice consiste à construire à partir de $\mathcal{T}_{\mathcal{I}}$, un arbre $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ dont les sommets sont des parties de $\mathcal{I} \times \mathcal{I}$ de la forme $\sigma \times \tau$. Ainsi, chaque sommet $\sigma \times \tau$ de l'arbre $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ peut être associé au bloc $M_{|\sigma \times \tau|}$ de la matrice M . Un tel arbre sera qualifié de Block Cluster Tree et la racine sera l'ensemble $\mathcal{I} \times \mathcal{I}$. Pour décrire la construction de $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$, il sera nécessaire d'introduire une fonction booléenne $f : \mathcal{P}(\mathcal{I}) \times \mathcal{P}(\mathcal{I}) \rightarrow \{0, 1\}$ qui sera notre critère d'admissibilité. Plus exactement, on dira qu'un bloc $\sigma \times \tau$ est admissible si $f(\sigma, \tau) = 1$. La construction de $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ est décrite par l'algorithme 4

où la notation $S(\sigma \times \tau)$ désignera les fils du sommet $\sigma \times \tau$ dans l'arbre $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ tandis que la notation $S(\sigma)$ désignera les fils du sommet σ dans l'arbre $\mathcal{T}_{\mathcal{I}}$.

Algorithme 4 CRÉATION DU BLOCK CLUSTER TREE

```

déf createClusterTree( $\sigma \in \mathcal{T}_{\mathcal{I}}, \tau \in \mathcal{T}_{\mathcal{I}}$ )
  si  $f(\sigma, \tau) == 1$  ou  $S(\sigma) == \emptyset$  ou  $S(\tau) == \emptyset$  alors
     $S(\sigma \times \tau) = \emptyset$ 
  sinon
     $S(\sigma \times \tau) = \{\sigma' \times \tau', \sigma' \in S(\sigma) \text{ et } \tau' \in S(\tau)\}$ 
    pour  $\sigma' \times \tau' \in S(\sigma \times \tau)$ 
      createClusterTree( $\sigma', \tau'$ )

```

Fin

Une fois que l'on obtient $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$, on remarque que si un sommet $\sigma \times \tau$ est admissible alors c'est une feuille. On espère de façon générale que les feuilles admissibles soient des blocs de grande taille et que N_{leaf} ait été choisi suffisamment petit pour que les feuilles non admissibles soient de petites taille. Finalement grâce à $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$, pour $\sigma \times \tau$ une feuille admissible, le bloc $M_{|\sigma \times \tau}$ de la matrice M subira une compression pour être au format AB^T .

Pour la fonction booléenne f , son choix dépend du type de matrice que l'on considère et aussi du type de compression des blocs admissibles. On se contentera donc d'exposer rapidement un critère d'admissibilité évoqué dans la thèse de Lizé. On définit pour $\sigma, \tau \subset \mathcal{I}$,

$$\text{diam}(\sigma) = \sqrt{\sum_{i=1}^d (x_{max, \sigma}^i - x_{min, \sigma}^i)^2} \quad (3.1)$$

$$d(\sigma, \tau) = \sqrt{\sum_{i=1}^d \max(0, x_{min, \tau}^i - x_{max, \sigma}^i)^2 + \max(0, x_{min, \sigma}^i - x_{max, \tau}^i)^2} \quad (3.2)$$

$\text{diam}(\sigma)$ désigne le diamètre de la boîte englobant les points $(x^j)_{j \in \sigma}$ et $d(\sigma, \tau)$ désigne la distance des faces les plus proches des deux boîtes englobantes associées à σ et à τ . Sous ces définitions, il est maintenant possible de définir pour $\eta \in \mathbb{R}_+^*$ un critère d'admissibilité f_η : pour $\sigma, \tau \subset \mathcal{I}$:

$$f_\eta(\sigma, \tau) = \begin{cases} 1 & \text{si } \min(\text{diam}(\sigma), \text{diam}(\tau)) < \eta \cdot d(\sigma, \tau) \\ 0 & \text{sinon} \end{cases} \quad (3.3)$$

η est parfois appelé facteur d'admissibilité.

Remarque 3.2.3. Les feuilles de $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ forment une partition de $\mathcal{I} \times \mathcal{I}$. La représentation en \mathcal{H} -matrice consiste alors à stocker la matrice M au niveau des feuilles de l'arbre $\mathcal{T}_{\mathcal{I} \times \mathcal{I}}$ où une feuille $\sigma \times \tau$ stockera aussi le sous-bloc $M_{|\sigma \times \tau}$ de la matrice M . Si de plus la feuille est admissible, alors on stockera A et B deux matrices issues de la compression et vérifiant $M_{|\sigma \times \tau} \approx AB^T$.

3.2.2 Compression des blocs admissibles

Si on considère $\sigma \times \tau$ un bloc admissible, alors on espère que l'on dispose de $k \in \mathbb{N}^*$ vraiment plus petit que $|\sigma|$ et $|\tau|$, de $A \in M_{|\sigma|,k}(\mathbb{R})$ et $B \in M_{|\tau|,k}(\mathbb{R})$ tels que $M_{|\sigma \times \tau} \approx AB^T$ au sens d'une norme matricielle (on ne considérera que la norme spectrale $\|\cdot\|_2$ et la norme de Frobenius $\|\cdot\|_F$). La représentation approchée de $M_{|\sigma \times \tau}$ par AB^T est qualifiée de $\mathcal{R}k$ -Matrice. Le théorème qui suit présente la meilleure façon d'approcher une matrice par une $\mathcal{R}k$ -Matrice (SVD ou décomposition en valeurs singulières).

Théorème 3.2.1. Soit $(p, q) \in (\mathbb{N}^*)^2$, $A \in M_{p,q}(\mathbb{R})$. On note $A = U\Sigma V^T$ une décomposition en valeurs singulières de A où $U \in M_{p,p}(\mathbb{R})$ et $V \in M_{q,q}(\mathbb{R})$ sont unitaires et $\Sigma \in M_{p,q}(\mathbb{R})$ est nulle sauf au niveau de sa diagonale et ses éléments diagonaux sont des réels positifs classés par ordre décroissant (ce sont les valeurs singulières).

Soit $k \in \llbracket 1; \min(p, q) \rrbracket$. On note $\text{diag}_k(\Sigma)$ la matrice Σ où pour i entier strictement plus grand que k , $\sigma_i = \Sigma_{i,i}$ a été remplacé par 0.

Alors si on note $\|\cdot\|$ la norme de Frobenius ou la norme spectrale, on a :

$$\|A - U\text{diag}_k(\Sigma)V^T\| = \min\{\|A - R\|, R \in M_{p,q}(\mathbb{R}) \text{ tel que } \text{rang}(R) \leq k\}$$

On a de plus :

$$\min\{\|A - R\|_2, R \in M_{p,q}(\mathbb{R}) \text{ tel que } \text{rang}(R) \leq k\} = \sigma_{k+1}^2 \quad (3.4)$$

$$\min\left\{\|A - R\|_F^2, R \in M_{p,q}(\mathbb{R}) \text{ tel que } \text{rang}(R) \leq k\right\} = \sum_{i=k+1}^{\min(p,q)} \sigma_i^2 \quad (3.5)$$

Remarque 3.2.4. Les formules (3.4) et (3.5) montrent que l'erreur d'approximation est complètement contrôlée par les valeurs singulières de A .

Remarque 3.2.5. On observe que $U\text{diag}_k(\Sigma)V^T = PQ^T$ où $P = U\text{diag}_k(\Sigma)$ et $Q = V$. D'où en plus d'être une $\mathcal{R}k$ -Matrice, $U\text{diag}_k(\Sigma)V^T$ est une approximation optimale de A par une matrice de rang au plus k . Rajoutons que si on note $U_k \in M_{p,k}(\mathbb{R})$ (resp. $V_k \in M_{q,k}(\mathbb{R})$) la matrice U (resp. V) dont on ne retient que les k premières colonnes, alors on a la représentation tronquée de

rang k , $U \text{diag}_k(\Sigma) V^T = U_k \text{diag}_k(\Sigma) V_k^T$, ce qui amène lorsque k est connu à remplacer U et V par U_k et V_k .

En pratique, pour quantifier l'erreur, on utilise la norme de Frobenius et on a une quantité $\epsilon \in \mathbb{R}_+^*$ qui est l'erreur absolue à ne pas dépasser. Ainsi si on reconsidère la matrice A du théorème précédent, la méthode SVD conditionnée par ϵ consiste à déterminer d'abord une décomposition en valeurs singulières (SVD) de la matrice A puis à choisir le rang k comme étant le plus petit entier entre 1 et $\min(p, q)$ tel que :

$$\|A - U \text{diag}_k(\Sigma) V^T\|_F = \sqrt{\sum_{i=k+1}^{\min(p,q)} \sigma_i^2} \leq \epsilon$$

Si les blocs admissibles sont compressés par la méthode SVD conditionnée par ϵ , on appelle alors ϵ epsilon d'assemblage.

Cependant effectuer la SVD de la matrice A admet une complexité en temps en $\mathcal{O}(pq^2 + p^2q)$. Donc compresser les blocs admissibles par SVD pour obtenir pour chaque bloc une représentation tronquée conditionnée par l'erreur absolue ϵ peut s'avérer coûteux. C'est pourquoi il est préférable d'opter pour un autre moyen de compression.

Les méthodes populaires de compression sont les méthodes ACA (Adaptive Cross Approximation). Ces méthodes consistent à approcher une matrice $A \in M_{p,q}(\mathbb{R})$ par des approximations successives de rang 1, $R_1 = C_1 D_1^T, R_2 = C_2 D_2^T, \dots$ où les C_i (resp. D_i) sont des vecteurs colonnes de taille p (resp. de taille q) et on cherche le plus petit entier k vérifiant un certain critère d'arrêt pouvant être par exemple

$$\|A - \sum_{i=1}^k R_i\|_F \leq \epsilon_1 \quad (3.6)$$

où $\epsilon_1 \in \mathbb{R}_+^*$ (appelé epsilon d'assemblage si on utilise ce critère pour compresser les blocs admissibles). L'approximation de A , \tilde{A} , est définie ainsi

$$\tilde{A} = \sum_{i=1}^k R_i$$

\tilde{A} est bien une $\mathcal{R}k$ -Matrice car si on pose

$$C = (C_1 \ C_2 \ \dots \ C_k) \text{ et } D = \begin{pmatrix} D_1 & 0_{M_{q,1}(\mathbb{R})} & \dots & 0_{M_{q,1}(\mathbb{R})} \\ 0_{M_{q,1}(\mathbb{R})} & D_2 & \dots & 0_{M_{q,1}(\mathbb{R})} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{M_{q,1}(\mathbb{R})} & 0_{M_{q,1}(\mathbb{R})} & \dots & D_k \end{pmatrix}$$

alors

$$\tilde{A} = CD^T$$

Rajoutons qu'afin de réduire le rang de la matrice \tilde{A} obtenue par une méthode ACA, il est possible d'appliquer une recompression de la matrice \tilde{A} par une variante de la SVD (voir les pages de 58 à 60 de [9]). Dans cette variante, on applique un moment donné la méthode SVD conditionnée par un réel strictement positif ϵ que l'on appelle epsilon de recompression.

Parmi les méthodes ACA, citons l'ACA pivotage total, l'ACA pivotage partiel, l'ACA+. Ces 3 méthodes sont évoquées dans la thèse de Lizé [9].

3.3 Complexité spatiale et temporelle

L'algorithme 2 version \mathcal{H} -matrice pour une réalisation nécessite plusieurs étapes : compresser en \mathcal{H} -matrice de la matrice de covariance $\Sigma \in M_r(\mathbb{R})$ où $r = dm$ (cette phase s'appelle aussi assemblage), obtenir L , la factorisation de Cholesky version \mathcal{H} -matrice de Σ , faire une réalisation y du vecteur gaussien de loi $\mathcal{N}(0, I_r)$, et enfin faire le produit matrice-vecteur Ly .

La connaissance de la complexité en temps et en espace de ces étapes permettrait d'obtenir celles de l'algorithme de simulation par \mathcal{H} -matrice et on s'attend à ce qu'une forte compression de la matrice de covariance réduise les coûts de temps et de mémoire. Cependant cette compression dépend entre autres du type de matrices que l'on considère et du découpage spatial. Par conséquent on ne peut donner de façon générale la complexité en temps et en espace de l'algorithme 2 version \mathcal{H} -matrice pour une réalisation. Cependant en se référant à [9], on peut espérer pour certaines matrices une complexité spatiale de l'algorithme de simulation en $O(r \log(r))$ et une complexité temporelle en $O(r \log^2(r))$.

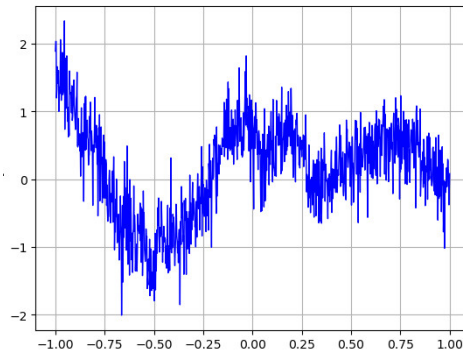


FIGURE 3.2 – Réalisation sur $[-1, 1]$ par méthode des \mathcal{H} -matrices

3.4 Erreur et convergence numérique

3.4.1 Paramétrage pour les tests

Pour les tests numériques, la méthode des \mathcal{H} -matrices nécessite d'abord l'information d'un maillage M et d'une fonction de covariance C . Afin de comparer cette méthode avec celle de Cholesky, on choisit que le choix des maillages pour les dimensions 1, 2 et 3 et celui des fonctions de covariance seront les mêmes que ceux décrits pour la méthode de Cholesky à la section 2.3 du chapitre 2. Cependant dans les sections précédentes on a vu que d'autres paramètres entraient en jeu, en raison de la compression en \mathcal{H} -matrice de la matrice de covariance $\Sigma \in M_{dm}(\mathbb{R})$. Pour que la compression soit possible, on associe à chaque indice $i \in \llbracket 1; dm \rrbracket$ un nœud du maillage M en suivant la logique évoquée à la remarque 3.2.1.

Ces nouveaux paramètres ont été fixés de façon identique pour tous les maillages et fonctions de covariance sélectionnés pour les tests : le paramètre N_{leaf} (voir la sous-sous-section 3.2.1.1) est fixé à 250 et le type de découpage hiérarchique de l'espace sera le découpage médian. Pour le découpage hiérarchique de la matrice (voir la sous-sous-section 3.2.1.2), le critère d'admissibilité (fonction booléenne) est la fonction définie par la formule (3.3) dont le facteur d'admissibilité η est fixé à 100. Pour la compression des blocs admissibles (voir la sous-section 3.2.2), on choisit la méthode ACA+ dont l'épsilon de recompression ϵ_1 est fixé à 10^{-7} (voir formule (3.6)). Les blocs admissibles subiront après la méthode ACA+, une autre recompression par la variante de la SVD où l'épsilon de recompression est fixé à 10^{-7} .

3.4.2 Benchmarks

Dimension	Nb de nœuds	Temps moyen d'une réalisation (en seconde)
1	10	0.00014s
1	100	0.00076s
1	1000	0.07200s
1	10000	0.55147s
2	9	0.00024s
2	100	0.00095s
2	961	0.08200s
2	10000	6.2282s
3	8	0.00035s
3	64	0.00063s
3	729	0.07902s
3	9261	60.024s

Dimension	Nb de nœuds	Nb de réalisations	erreur L^2
1	10	250	0.16838
1	10	500	0.11806
1	10	750	0.10538
1	10	1000	0.09248
1	100	250	0.28641
1	100	500	0.19507
1	100	750	0.16492
1	100	1000	0.13788
1	1000	250	0.29194
1	1000	500	0.22217
1	1000	750	0.16064
1	1000	1000	0.14545
2	9	250	0.15007
2	9	500	0.10969
2	9	750	0.09103
2	9	1000	0.07789
2	100	250	0.44965
2	100	500	0.31608
2	100	750	0.25780
2	100	1000	0.22462
2	961	250	0.92465
2	961	500	0.64501
2	961	750	0.52958
2	961	1000	0.45912
3	8	250	0.15190
3	8	500	0.09414
3	8	750	0.08731
3	8	1000	0.07096
3	64	250	0.36302
3	64	500	0.25823
3	64	750	0.21156
3	64	1000	0.18688
3	729	250	1.1965
3	729	500	0.84435
3	729	750	0.69320
3	729	1000	0.59755

Chapitre 4

Méthode de Galli-Gao-Gibbs

Cette méthode consiste à approcher des réalisations de la loi $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$ à l'aide d'un échantillonnage de Gibbs (méthode de Monte-Carlo par chaîne de Markov). Cependant l'échantillonnage de Gibbs en lui-même ne cherchera pas à approcher la loi $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$ mais la loi $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma^{-1})$. Puis toute réalisation approchée de $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma^{-1})$ subira une transformation linéaire dont le résultat sera une réalisation approchée de $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$.

4.1 Échantillonnage de Gibbs

Décrivons d'abord ce en quoi consiste l'échantillonnage de Gibbs.

Considérons r un entier non nul, $Y = (Y_1, \dots, Y_r)$ un vecteur aléatoire à valeurs dans \mathbb{R}^r et $\lambda_1, \dots, \lambda_r$ r mesures positives sur $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Notons $\lambda = \lambda_1 \otimes \dots \otimes \lambda_r$ leur mesure-produit sur $(\mathbb{R}^r, \mathcal{B}(\mathbb{R}^r))$.

On suppose que \mathbb{P}_Y la loi de Y est à densité par rapport à la mesure λ . Nommons $\pi : \mathbb{R}^r \rightarrow \mathbb{R}_+$ cette densité. On admet aussi que π est strictement positive.

Imposons les notations suivantes :

- pour $i \in \llbracket 1; r \rrbracket$, $Y^i = (Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_r)$
- pour $i \in \llbracket 1; r \rrbracket$ et $y = (y_1, \dots, y_r) \in \mathbb{R}^r$, on note $y^i = (y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_r)$ et $\pi_i(\cdot | y^i)$ la loi conditionnelle de Y_i sachant $Y^i = y^i$

Alors la mesure de probabilité $\pi_i(\cdot, y^i)$ admet une densité conditionnelle $\tilde{\pi}_i(\cdot, y^i)$ par rapport à la mesure λ_i . Cette densité peut être décrite à l'aide de π et de la densité π_{Y^i} de \mathbb{P}_{Y^i} par rapport à la mesure $\lambda^i = \lambda_1 \otimes \dots \otimes \lambda_{i-1} \otimes \lambda_{i+1} \otimes \dots \otimes \lambda_r$. D'ailleurs dans la littérature mathématique, $\pi_i(\cdot, y^i)$ et $\tilde{\pi}_i(\cdot, y^i)$ sont souvent confondues dans les notations.

Remarque 4.1.1. Pour tout $x = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_r) \in \mathbb{R}^{r-1}$ et pour $u \in \mathbb{R}$

$$\pi_{Y^i}(x) = \int_{\mathbb{R}} \pi(x_1, \dots, x_{i-1}, z, x_{i+1}, \dots, x_r) d\lambda_i(z) > 0$$

$$\tilde{\pi}_i(u|x) = \frac{\pi(x_1, \dots, x_{i-1}, u, x_{i+1}, \dots, x_r)}{\pi_{Y^i}(x)}.$$

Dans ce cadre, il est possible de décrire l'échantillonnage de Gibbs par l'algorithme suivant :

Algorithme 5 ÉCHANTILLONNAGE DE GIBBS

Entrée(s) x liste nulle de longueur r ,

$(\pi_i(\cdot|\cdot))_{i \in \{1, r\}}$ les r lois conditionnelles,

N un entier non nul (nombre maximal d'itérations)

Début

pour $a := 1$ à N **faire**

on choisit $i \in \llbracket 1; r \rrbracket$ selon une loi uniforme

on tire c un réel selon la loi $\pi_i(\cdot|x^i)$

$x[i] = c$

fin du pour

Fin

Sortie(s) x

Cet échantillonnage produit une chaîne de Markov $(X_i)_{i \in \mathbb{N}}$ à valeurs dans \mathbb{R}^r où l'indice i présente le nombre d'itérations de l'algorithme. En particulier X_0 est constant de valeur $0_{\mathbb{R}^r}$. Cette chaîne de Markov vérifie de multiples propriétés qui sont évoquées dans le chapitre 4 de la référence [3]. Évoquons surtout celle-ci :

Propriété 4.1.1. La stricte positivité de la densité π garantit que la chaîne de Markov $(X_i)_{i \in \mathbb{N}}$ converge en loi vers Y .

Donc plus on augmente le nombre d'itérations de l'échantillonnage, plus le résultat de l'algorithme approche une réalisation de Y .

4.2 Cas gaussien

4.2.1 Méthode de Gibbs

Suite à la présentation de l'échantillonnage de Gibbs, on s'intéresse maintenant à la simulation du vecteur gaussien $Z = X_M \sim \mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$ où $r = dm$. On notera que l'on est dans le cadre où on peut faire un tel échantillonnage. En effet Z est à densité par rapport à la mesure de Lebesgue de \mathbb{R}^r , $\lambda^r = \lambda \otimes \dots \otimes \lambda$, où λ est la mesure de Lebesgue de \mathbb{R} . Si on note π cette densité, alors pour $x \in \mathbb{R}^r$:

$$\pi(x) = \frac{1}{(2\pi)^{r/2} \det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2} x^T \Sigma^{-1} x\right) > 0$$

Il est donc possible de simuler Z par un échantillonnage de Gibbs. Cependant un problème se manifeste : si on utilise les notations de la section précédente, il faut déterminer pour $i \in \llbracket 1; r \rrbracket$ et $x = (x_1, \dots, x_r) \in \mathbb{R}^r$, $\pi_i(\cdot | x^i)$ la loi conditionnelle de Z_i sachant $Z^i = x^i$. Dans le cas gaussien dans lequel nous sommes, une description explicite de $\pi_i(\cdot | x^i)$ est possible comme le confirme la référence [8]. On a que $\pi_i(\cdot | x^i)$ est la loi gaussienne univariée $\mathcal{N}(z_{i, \Sigma^{-1}}(x), \sigma_{i, \Sigma^{-1}}^2)$ où :

$$z_{i, \Sigma^{-1}}(x) = \sum_{j=1, j \neq i}^r -\frac{(\Sigma^{-1})_{i,j}}{(\Sigma^{-1})_{i,i}} x_j \quad \sigma_{i, \Sigma^{-1}}^2 = \frac{1}{(\Sigma^{-1})_{i,i}}$$

D'où l'échantillonnage de Gibbs suivant :

Algorithme 6 ÉCHANTILLONNAGE DE GIBBS : CAS $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$

Entrée(s) x liste nulle de longueur r , N un entier non nul

Début

pour $a := 1$ **à** N **faire**

on choisit $i \in \llbracket 1; r \rrbracket$ selon une loi uniforme

on tire c un réel selon la loi $\mathcal{N}(z_{i, \Sigma^{-1}}(x), \sigma_{i, \Sigma^{-1}}^2)$

$x[i] = c$

fin du pour

Fin

Sortie(s) x

4.2.2 Méthode de Galli-Gao-Gibbs

On remarque que dans l'algorithme 6, il est nécessaire de connaître l'inverse de Σ , ce qui implique de lourds calculs si r est grand. Par conséquent, on peut se demander s'il serait possible de contourner cette difficulté en n'utilisant que Σ . Dans l'article de Galli-Gao [4], une solution est proposée et est fondée sur la remarque suivante : si on considère le vecteur aléatoire $Y = \Sigma^{-1}Z$ alors $Y \sim \mathcal{N}(0_{\mathbb{R}^r}, \Sigma^{-1})$. Donc comme l'inverse de Σ^{-1} vaut Σ , simuler Y par un échantillonnage de Gibbs ne nécessite que la connaissance de Σ . De plus $\Sigma Y = Z \sim \mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$. D'où la possibilité de simuler Z en ne connaissant que Σ : on fait un échantillonnage de Gibbs (EG) simulant une réalisation y de Y puis on considère le vecteur Σy afin d'obtenir une réalisation de Z .

Ce choix de simulation s'avère pertinent mathématiquement. En effet si on note $(Y_N)_{N \in \mathbb{N}}$ la chaîne de Markov issue de l'EG simulant $Y \sim \mathcal{N}(0_{\mathbb{R}^r}, \Sigma^{-1})$, alors par la propriété 4.1.1 de la section 4.1, la suite $(Y_N)_{N \in \mathbb{N}}$ converge en loi vers Y et de cette information on peut déduire que $(Z_N)_{N \in \mathbb{N}} = (\Sigma Y_N)_{N \in \mathbb{N}}$ converge en loi vers Z .

On pourrait se dire qu'on a notre nouvel algorithme de simulation de $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$. Cependant on peut faire la remarque suivante : choisissons un entier naturel N et notons $y(N)$ le vecteur de \mathbb{R}^r obtenu après la N -ième itération de l'EG pour simuler une réalisation de $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma^{-1})$. Après la $(N+1)$ -ième itération, grâce à un entier a choisi au hasard dans $\llbracket 1; r \rrbracket$, on obtient un vecteur $y(N+1)$ qui est le vecteur $y(N)$ mais où la a -ième composante de $y(N)$ a été remplacée par un réel v selon la loi $\mathcal{N}(z_{a,\Sigma}(y(N)), \sigma_{a,\Sigma}^2)$. Or pour simuler $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$, il est plus intéressant de considérer les vecteurs $z(N) = \Sigma y(N)$ et $z(N+1) = \Sigma y(N+1)$. Établissons alors une relation entre $z(N)$ et $z(N+1)$ qui ne nécessite pas de connaître $y(N)$ et $y(N+1)$ en s'inspirant de la référence [8] :

Pour simuler $\mathcal{N}(z_{a,\Sigma}(y(N)), \sigma_{a,\Sigma}^2)$, il est possible de considérer la variable aléatoire réelle $V = z_{a,\Sigma}(y(N)) + \sigma_{a,\Sigma}U$ où $U \sim \mathcal{N}(0, 1)$. On considèrera alors u le réel tel que :

$$\begin{aligned} v &= z_{a,\Sigma}(y(N)) + \sigma_{a,\Sigma}u \\ &= - \sum_{j=1, j \neq a}^r \frac{\Sigma_{a,j}}{\Sigma_{a,a}} y(N)_j + \frac{1}{\sqrt{\Sigma_{a,a}}} u \end{aligned}$$

On obtient alors pour $k \in \llbracket 1; r \rrbracket$:

$$\begin{aligned}
 z(N+1)_k &= \sum_{j=1, j \neq a}^r \Sigma_{k,j} y(N+1)_j + \Sigma_{k,a} y(N+1)_a \\
 &= \sum_{j=1, j \neq a}^r \Sigma_{k,j} y(N)_j + \Sigma_{k,a} v \\
 &= \sum_{j=1, j \neq a}^r \Sigma_{k,j} y(N)_j + \Sigma_{k,a} \left(- \sum_{j=1, j \neq a}^r \frac{\Sigma_{a,j}}{\Sigma_{a,a}} y(N)_j + \frac{1}{\sqrt{\Sigma_{a,a}}} u \right) \\
 &= \sum_{j=1}^r \Sigma_{k,j} y(N)_j - \Sigma_{k,a} y(N)_a + \Sigma_{k,a} \left(- \sum_{j=1, j \neq a}^r \frac{\Sigma_{a,j}}{\Sigma_{a,a}} y(N)_j + \frac{1}{\sqrt{\Sigma_{a,a}}} u \right) \\
 &= \sum_{j=1}^r \Sigma_{k,j} y(N)_j + \frac{\Sigma_{k,a}}{\Sigma_{a,a}} \left(- \Sigma_{a,a} y(N)_a - \sum_{j=1, j \neq a}^r \Sigma_{a,j} y(N)_j + \sqrt{\Sigma_{a,a}} u \right) \\
 &= z(N)_k + \frac{\Sigma_{k,a}}{\Sigma_{a,a}} \left(- \sum_{j=1}^r \Sigma_{a,j} y(N)_j + \sqrt{\Sigma_{a,a}} u \right) \\
 &= z(N)_k + \frac{\Sigma_{k,a}}{\Sigma_{a,a}} \left(- z(N)_a + \sqrt{\Sigma_{a,a}} u \right)
 \end{aligned}$$

Voici la relation recherchée entre $z(N)$ et $z(N+1)$. On observe aussi que $z(N+1)_a = \sqrt{\Sigma_{a,a}} u$ et que par la relation entre u et v , la valeur $z(N+1)_a$ est une réalisation de la gaussienne $\mathcal{N}(0, \Sigma_{a,a})$.

Donc, pour récapituler, on obtient $z(N+1)$ à partir de $z(N)$ et de a de la façon suivante :

- on tire un réel c selon la gaussienne $\mathcal{N}(0, \Sigma_{a,a})$
- $z(N+1)_a = c$
- pour $k \in \llbracket 1; r \rrbracket$, $k \neq a$, $z(N+1)_k = z(N)_k + \frac{\Sigma_{k,a}}{\Sigma_{a,a}} (z(N+1)_a - z(N)_a)$

Finalement si on note N_{max} le nombre maximal d'itérations pour l'EG, $z(N_{max}) = \Sigma y(N_{max})$ simule une réalisation de $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$ et on vient de prouver que $z(N_{max})$ peut être obtenu sans la connaissance des $(y(i))_{i \in \llbracket 1; N_{max} \rrbracket}$ issue de l'EG mais par une relation récursive sur les $(z(i))_{i \in \llbracket 1; N_{max} \rrbracket}$. Ceci conduit à l'algorithme 7 pour simuler $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$.

Algorithme 7 MÉTHODE DE GALLI-GAO-GIBBS : CAS $\mathcal{N}(0_{\mathbb{R}^r}, \Sigma)$

Entrée(s) z^n, z^c deux listes nulles de longueur r ,
 N un entier non nul (nombre d'itérations)

Début

pour $i := 1$ à N **faire**

on choisit $a \in \llbracket 1; r \rrbracket$ selon une loi uniforme

on tire c un réel selon la loi $\mathcal{N}(0, \Sigma_{a,a})$

$z^n[a] = c$

pour $k := 1$ à r tel que $k \neq a$ **faire**

$z^n[k] = z^c[k] + \frac{\Sigma_{k,a}}{\Sigma_{a,a}}(z^n[a] - z^c[a])$

fin du pour

$z^c = z^n$

fin du pour

Fin

Sortie(s) z^c

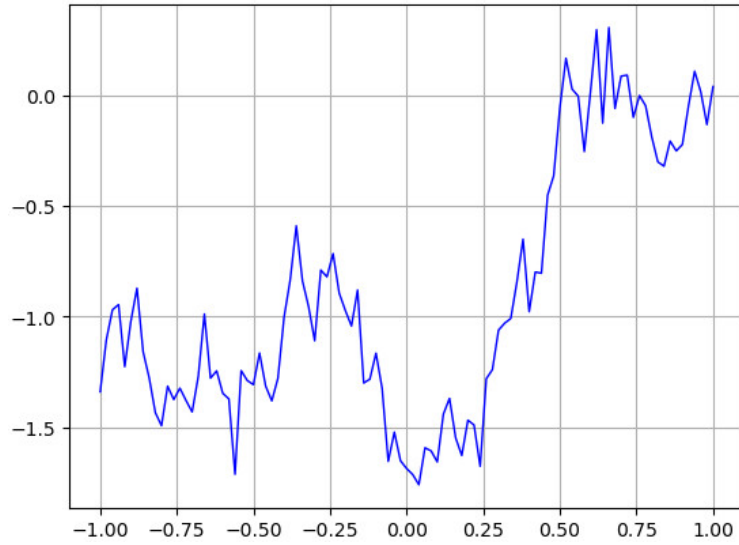


FIGURE 4.1 – Réalisation sur $[-1, 1]$ par Galli-Gao-Gibbs

Remarque 4.2.1. Dans OpenTURNS, l'algorithme se veut un peu différent en décidant pour une itération non pas de choisir un élément $a \in \llbracket 1; r \rrbracket$ selon une loi uniforme mais de tirer une permutation σ de $\llbracket 1; r \rrbracket$ selon une loi uniforme puis de faire r itérations où a vaut d'abord $\sigma(1)$, puis $\sigma(2)$ jusqu'à $\sigma(r)$. Donc dans OpenTURNS, le nombre d'itérations N_{it} mis en paramètre correspond dans les faits à $r \cdot N_{it}$ itérations où r est la taille du vecteur gaussien à simuler.

4.3 Complexité algorithmique

L'algorithme 7 admet une complexité en temps en $O(Nr)$ et une complexité en mémoire en $O(r)$ (le stockage par des listes z^n, z^c pour la simulation). Par contre, pour l'algorithme 7 version OpenTURNS, en se fiant à la remarque 4.2.1, bien que la complexité en mémoire reste inchangée, la complexité en temps est en $O(N_{it}r^2)$ (pour une itération, lorsque a parcourt les valeurs de la permutation σ , $O(r)$ opérations sont effectuées à chaque fois).

4.4 Erreur et convergence numérique

4.4.1 Dimension $n=1$

Comme pour la méthode de Cholesky et celle des \mathcal{H} -matrices, pour simuler on a besoin d'un maillage M et d'une fonction de covariance C pour déduire Σ , la matrice de covariance du problème Σ . Le choix des maillages et des fonctions de covariance sera celui des maillages de la dimension $n = 1$ évoqués pour la méthode de Cholesky (voir la sous-section 2.3.1 du chapitre 2) et le nombre m de nœuds dans les maillages ne variera plus entre 10 et 10000 mais plutôt entre 10 et 100 (pour des raisons de temps de calcul). Pour appliquer la méthode de Galli-Gao-Gibbs, il faut aussi calibrer le nombre d'itérations N_{it} évoqué à la remarque 4.2.1. On décidera de le faire varier entre 10 et 300 pour les tests.

4.4.2 Benckmarks

Dimension	Nb de nœuds	N_{it}	Temps moyen d'une réalisation (en secondes)
1	10	100	0.00637s
1	10	200	0.01468s
1	10	300	0.02217s
1	100	100	0.15972s
1	100	200	0.34042s
1	100	300	0.47059s

Dimension	Nb de nœuds	N_{it}	Nb de réalisations	erreur L^2
1	10	10	250	0.18609
1	10	10	500	0.11308
1	10	10	750	0.07441
1	10	10	1000	0.07942
1	10	50	250	0.17034
1	10	50	500	0.12572
1	10	50	750	0.08582
1	10	50	1000	0.08190
1	10	100	250	0.14295
1	10	100	500	0.10940
1	10	100	750	0.09015
1	10	100	1000	0.06226
1	100	10	250	0.25851
1	100	10	500	0.19295
1	100	10	750	0.15731
1	100	10	1000	0.13206
1	100	50	250	0.27869
1	100	50	500	0.18906
1	100	50	750	0.16029
1	100	50	1000	0.13728
1	100	100	250	0.30028
1	100	100	500	0.18475
1	100	100	750	0.14346
1	100	100	1000	0.13889

Chapitre 5

Simulation par méthode spectrale

La méthode qui va être présentée est fondée essentiellement sur un cours de l'école Polytechnique (voir [2]). Le lecteur est invité à lire l'annexe A afin de mieux saisir les objets mathématiques utilisés dans ce chapitre.

Soit (Ω, F, \mathbb{P}) un espace probabilisé, n et d deux entiers naturels non nuls et $X = (X_1, \dots, X_d) : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^d$ un champ aléatoire gaussien centré, continue en moyenne quadratique et faiblement stationnaire d'ordre 2, admettant pour mesure spectrale M_X . On supposera que M_X admet une densité spectrale $S_X : \mathbb{R}^n \rightarrow M_d(\mathbb{R})$.

On va chercher à simuler X sur un domaine appelé le domaine de simulation $\overline{T} = \overline{T}_1 \times \dots \times \overline{T}_n$ un pavé de \mathbb{R}^n où $\overline{T}_j = [0, T_j]$, $T_j \in \mathbb{R}_+^*$. Une fois ce domaine défini, on discrétisera \overline{T} en une famille finie de points et on simulera X en ces points que l'on appellera les points de simulation. À la fin, on obtiendra un champ aléatoire gaussien discret qui approchera X sur \overline{T} .

5.1 Hypothèses sur la densité spectrale

Commençons en imposant trois hypothèses sur la densité spectrale S_X . Les deux premières hypothèses permettront de justifier mathématiquement le choix de la discrétisation du domaine de simulation et la dernière permettra de construire l'algorithme de simulation de la méthode spectrale.

Hypothèse 5.1.1. S_X est à support compact dans le pavé $\overline{\Omega} = \overline{\Omega}_1 \times \dots \times \overline{\Omega}_n$ où $\forall j \in \llbracket 1; n \rrbracket$, $\overline{\Omega}_j = [-\Omega_{cj}, \Omega_{cj}]$, $\Omega_{cj} \in \mathbb{R}_+^*$

Hypothèse 5.1.2. $tr(S_X)$ est bornée (où $tr : M_d(\mathbb{C}) \rightarrow \mathbb{C}$ est la fonction trace)

Hypothèse 5.1.3. $\forall \omega \in \overline{\Omega} \setminus \partial\overline{\Omega}$, $S(\omega)$ est de rang d (donc $S(\omega)$ est hermitienne définie positive)

En général, S_X n'est pas forcément à support compact. Néanmoins :

- $\forall t \in \mathbb{R}^n, \mathbb{E}(\|X(t)\|_2^2) = \int_{\mathbb{R}^n} \text{tr}(S_X(\omega)) d\omega < \infty$
- en pratique, $S_X(\omega)$ tend rapidement vers $0_{M_d(\mathbb{R})}$ quand $\|\omega\| \rightarrow \infty$

Dans ces conditions on peut espérer trouver, pour $\epsilon \in \mathbb{R}_+^*$, arbitrairement petit, un pavé pas trop large $\overline{\Omega}^\epsilon = \overline{\Omega}_1^\epsilon \times \cdots \times \overline{\Omega}_n^\epsilon$ tel que :

- $\forall j \in \llbracket 1; n \rrbracket, \overline{\Omega}_j^\epsilon = [-\Omega_{cj}^\epsilon, \Omega_{cj}^\epsilon], \Omega_{cj}^\epsilon \in \mathbb{R}_+^*$
- $\int_{\mathbb{R}^n \setminus \overline{\Omega}^\epsilon} \text{tr}(S_X(\omega)) d\omega \leq (\int_{\mathbb{R}^n} \text{tr}(S_X(\omega)) d\omega) \cdot \epsilon$

Dans une telle configuration, on serait alors tenté d'approcher S_X par $\mathbb{1}_{\overline{\Omega}^\epsilon} S_X$ pour ϵ assez petit. D'où l'idée de chercher non pas à simuler le champ aléatoire X directement mais à l'approcher par le champ aléatoire X^ϵ d'ordre 2, à valeurs dans \mathbb{R}^d , centré, gaussien, continue en moyenne quadratique, faiblement stationnaire d'ordre 2 et de densité spectrale $\mathbb{1}_{\overline{\Omega}^\epsilon} S_X$ (l'existence d'un champ gaussien à valeurs dans \mathbb{R}^d de densité spectrale $\mathbb{1}_{\overline{\Omega}^\epsilon} S_X$ peut se justifier mais on n'insistera pas sur ce point). On revient ainsi au cas de S_X à support compact.

5.2 Factorisation de Cholesky

Pour ω dans $\overline{\Omega} \setminus \partial\overline{\Omega}$, $S_X(\omega)$ est hermitienne définie positive, donc admet une décomposition de Cholesky $S_X(\omega) = \mathcal{H}(\omega)\mathcal{H}(\omega)^*$ où $\mathcal{H}(\omega)$ est une matrice triangulaire inférieure. Une décomposition (non unique) de Cholesky peut s'obtenir à l'aide la méthode de Cholesky généralisée aux matrices hermitiennes définies-positives.

5.3 Subdivision du domaine spectral

Soit N_1, \dots, N_n , n entiers naturels non nuls. Imposons d'abord un certain nombre de notations :

- $N = N_1 \times \cdots \times N_n$
- pour $j \in \llbracket 1; N \rrbracket$, $B_{N_j} = \llbracket 1; N_j \rrbracket$
- $B_N = B_{N_1} \times \cdots \times B_{N_n}$ (produit cartésien)

Une arête $[-\Omega_{cj}, \Omega_{cj}]$ sera subdivisée en N_j intervalles réguliers $M_{j,1}, \dots, M_{j,N_j}$ de longueur

$$\Delta\omega_j = \frac{2\Omega_{cj}}{N_j} \quad (5.1)$$

et de centres respectifs $\omega_{j,1}, \dots, \omega_{j,N_j}$.

Hypothèse 5.3.1. Pour la suite, on supposera que les N_j sont des multiples de 2. De cette façon, la subdivision $(M_k)_{k \in B_N}$ de $\overline{\Omega}$ est symétrique par rapport à l'origine (i.e si \mathcal{S} est la symétrie de \mathbb{R}^n qui à x associe $-x$, alors $\mathcal{S}(\overline{\Omega}) = \overline{\Omega}$ et pour tout $k \in B_N$, il existe un unique $j \in B_N$ tel que $j \neq k$ et $\mathcal{S}(M_k) = M_j$ (on a même que $\omega_j = -\omega_k$)) et la famille des centres $(\omega_k)_{k \in B_N}$ ne contient pas l'origine $(0, \dots, 0)$.

5.4 Subdivision du domaine de simulation

On pose pour $j \in \llbracket 1; n \rrbracket$, $f_{max,j} = \frac{\Omega_{cj}}{2\pi}$. Il s'agit de la fréquence maximale sur l'arête spectrale $[-\Omega_{cj}, \Omega_{cj}]$.

On obtient par le théorème d'échantillonnage de Shannon, que pour $j \in \llbracket 1; n \rrbracket$ la fréquence d'échantillonnage $f_{e,j}$ minimale requise pour représenter la densité spectrale S_X sur le $j^{\text{ème}}$ axe de coordonnées vaut $2f_{max,j}$. Le pas d'échantillonnage Δt_j se définit alors ainsi :

$$\forall j \in \llbracket 1; n \rrbracket, \Delta t_j = \frac{1}{f_{e,j}} = \frac{\pi}{\Omega_{cj}} \quad (5.3)$$

On déduit alors la relation : pour $j \in \llbracket 1; n \rrbracket$, $\Delta t_j \Delta \omega_j = \frac{2\pi}{N_j}$.

Donc une fois que la discrétisation (N_1, \dots, N_n) est donnée, le pas de pulsation $\Delta \omega_j$ et le pas de temps Δt_j ne peuvent être choisis indépendamment.

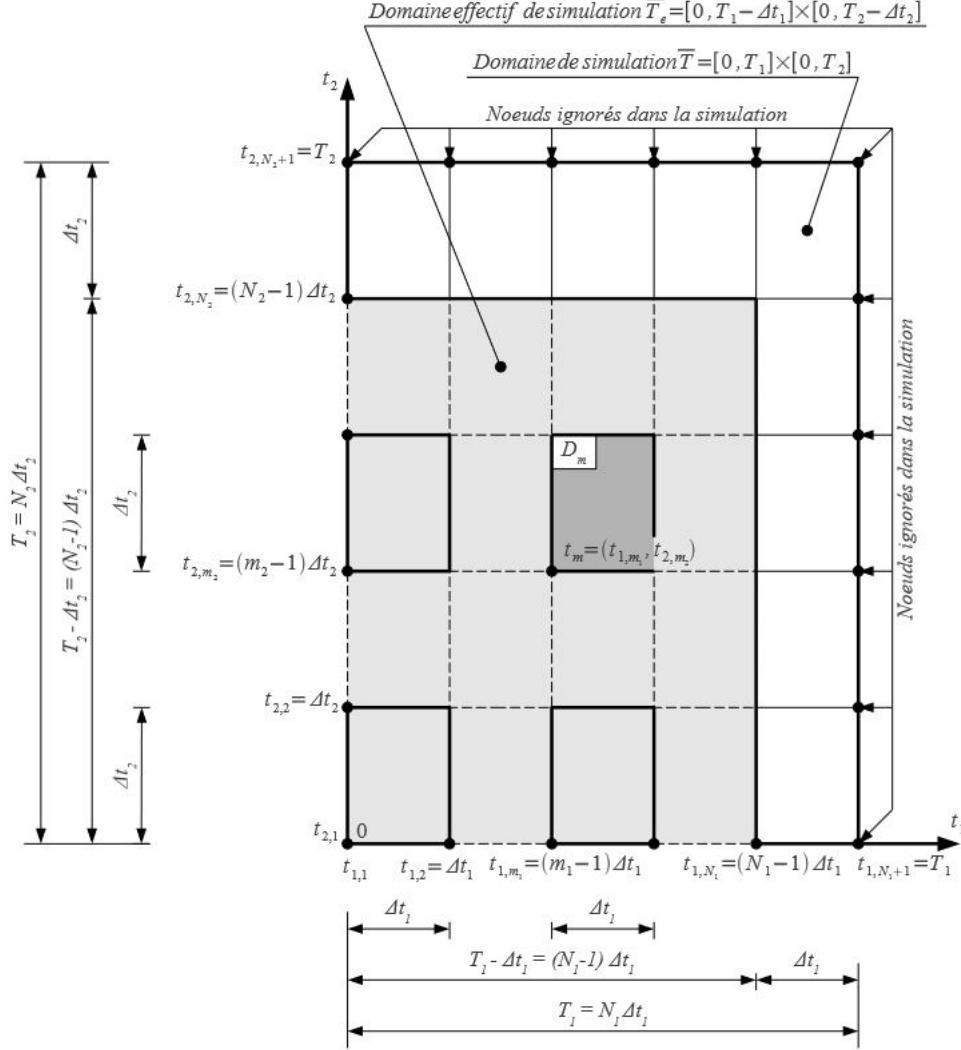
Par ailleurs sous les hypothèses 5.1.1 et 5.1.2, un théorème de Shannon plus fort garantit que la connaissance de $(X(k_1 \Delta t_1, \dots, k_n \Delta t_n))_{(k_1, \dots, k_n) \in \mathbb{Z}^n}$ suffit pour déduire la loi du champ X (voir page 446 de [12]).

Par conséquent pour simuler un champ aléatoire approchant X , on pourra plutôt chercher à simuler un nombre fini de variables aléatoires de la famille $(X(k_1 \Delta t_1, \dots, k_n \Delta t_n))_{(k_1, \dots, k_n) \in \mathbb{Z}^n}$. Explicitons lesquelles seront sélectionnées.

Posons pour $j \in \llbracket 1; n \rrbracket$, $m_j \in \llbracket 1; N_j \rrbracket$:

- $t_{j,m_j} = (m_j - 1)\Delta t_j$
- $T_j = N_j \Delta t_j$
- $\overline{T}_j = [0, T_j]$
- $\overline{T} = \overline{T}_1 \times \dots \times \overline{T}_n$ (domaine de simulation)
- pour $m \in B_N$, $t_m = (t_{1,m_1}, \dots, t_{n,m_n}) \in \overline{T}$ (points de simulation)

Avec ces notations, on peut maintenant dire que le but est de simuler la famille de vecteurs aléatoires $(X(t_m))_{m \in B_N}$ dont les éléments sont bien dans la famille $(X(k_1 \Delta t_1, \dots, k_n \Delta t_n))_{(k_1, \dots, k_n) \in \mathbb{Z}^n}$.


 FIGURE 5.2 – Subdivision du domaine de simulation en dimension $n = 2$

5.5 Approximation du champ gaussien X

X est un champ d'ordre 2, centré, continue en moyenne quadratique et faiblement stationnaire d'ordre 2. On peut alors utiliser le théorème A.4.1 de l'annexe A : on dispose de ϕ_X , la mesure stochastique associée à X et issue de la mesure stochastique orthogonale centrée (m.s.o.c.) I_X de mesure de base la mesure spectrale de X , M_X , telle que $\forall t \in \mathbb{R}^n$:

$$X(t, \cdot) = \int_{\mathbb{R}^n} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega) = \int_{\overline{\Omega}} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega) \quad (5.4)$$

Remarque 5.5.1. La dernière égalité de la relation (5.4) se justifie par le support compact de S_X dans $\overline{\Omega}$. En effet comme $M_X = S_X(\omega)d\omega$, on obtient que pour toute fonction f dans $L^2(\mathbb{R}^n, M_X)$, $\|M_X\|$ -presque partout, $f = \mathbf{1}_{\overline{\Omega}}f$ (voir la définition A.2.2).

Or $\overline{\Omega}$ a été subdivisé par les mailles M_k (voir la relation (5.2)) et ces mailles sont à deux à deux disjointes (à un ensemble Lebesgue-négligeable près). D'où, par linéarité de l'intégrale de Wiener et comme X est à valeurs dans \mathbb{R}^d , pour $t \in \mathbb{R}^n$:

$$\begin{aligned} X(t, \cdot) &= \sum_{k \in B_N} \int_{M_k} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega) \\ &= \Re \left(\sum_{k \in B_N} \int_{M_k} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega) \right) \end{aligned} \quad (5.5)$$

À partir de ce résultat, construisons peu à peu une approximation du processus X . On va d'abord faire l'approximation suivante :

$\forall t \in \mathbb{R}^n, \forall k \in B_N,$

$$\begin{aligned} \int_{M_k} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega) &\approx \int_{M_k} \exp(i \langle t, \omega_k \rangle_n) \phi_X(d\omega) \\ &= \exp(i \langle t, \omega_k \rangle_n) \phi_X(M_k) \end{aligned} \quad (5.6)$$

Remarque 5.5.2. On peut y voir une quadrature par méthode du point milieu généralisée.

On obtient alors la première approximation de X en posant pour $t \in \mathbb{R}^n$:

$$\begin{aligned} Y(t, \cdot) &= \Re \left(\sum_{k \in B_N} \exp(i \langle t, \omega_k \rangle_n) \phi_X(M_k) \right) \\ &= \Re \left(I_X \left(\sum_{k \in B_N} \exp(i \langle t, \omega_k \rangle_n) \mathbf{1}_{M_k} \right) \right) \end{aligned}$$

Pointons les propriétés suivantes (se justifiant à l'aide de la section A.4 de l'annexe A) :

1. Comme X est gaussien, les $\phi_X(M_k)$ sont des vecteurs gaussiens centrés de matrice de covariance complexe $\mathbb{E}(\phi_X(M_k) \phi_X(M_k)^*) = M_X(M_k) = \int_{M_k} S_X(\omega) d\omega$
2. Pour $(k, j) \in (B_N)^2$, tel que $k \neq j$,
 $\mathbb{E}(\phi_X(M_k) \phi_X(M_j)^*) = M_X(M_k \cap M_j) = 0_{M_d(\mathbb{C})}$.

3. Pour $(k, j) \in (B_N)^2$, $\mathbb{E}(\phi_X(M_k)\phi_X(M_j)^\top) = M_X(M_k \cap -(M_j))$
4. Y est un champ gaussien centré

Or par la symétrie de la subdivision $(M_k)_{k \in B_N}$ de $\overline{\Omega}$ (voir l'hypothèse 5.3.1), on a que pour tout $k \in B_N$, il existe un unique $j_k \in B_N$ différent de k tel que $M_{j_k} = -M_k$. D'où :

$$\forall (k, l) \in (B_N)^2, \mathbb{E}(\phi_X(M_k)\phi_X(M_l)^\top) = \begin{cases} M_X(M_k) & \text{si } l = j_k \\ 0_{M_d(\mathbb{C})} & \text{sinon} \end{cases} \quad (5.7)$$

Y est un champ gaussien centré donc Y est caractérisé par sa fonction de covariance que l'on note $C_Y : \mathbb{R}^n \times \mathbb{R}^n \rightarrow M_d(\mathbb{R})$. Par le calcul on obtient pour $(s, t) \in (\mathbb{R}^n)^2$:

$$C_Y(s, t) = \mathbb{E}(Y(s)Y(t)^\top) = \Re \left(\sum_{k \in B_N} e^{i\langle t-s, \omega_k \rangle_n} M_X(M_k) \right) \quad (5.8)$$

Ce résultat se fonde sur les propriétés 1, 2, 3, la formule (5.7) et sur la propriété suivante :

$$\forall (A, B) \in M_{d,1}(\mathbb{C})^2, \Re(A)\Re(B)^\top = \frac{1}{2}(\Re(AB^\top) + \Re(AB^*))$$

Le souci devient alors les $(M_X(M_k))$ qui ont le défaut d'être définis par une intégrale. Cependant, en posant $\Delta\omega = \Delta\omega_1 \dots \Delta\omega_n$, on peut approcher les $M_X(M_k)$ ainsi : $M_X(M_k) \approx \Delta\omega S_X(\omega_k)$ (encore une quadrature par méthode du point milieu généralisée).

On veut alors approcher le champ gaussien centré Y par un champ gaussien centré dont la fonction de covariance vaut :

$$K(s, t) = \Delta\omega \Re \left(\sum_{k \in B_N} e^{i\langle t-s, \omega_k \rangle_n} S_X(\omega_k) \right) \quad (5.9)$$

Or on peut montrer que le champ X_N suivant est à valeurs dans \mathbb{R}^d , gaussien, centré et que sa fonction de covariance vaut bien K :

$$\forall t \in \mathbb{R}^n, X_N(t, \cdot) = \Re \left(\sum_{k \in B_N} \sqrt{\Delta\omega} e^{i\langle t, \omega_k \rangle_n} \mathcal{H}(\omega_k) V_k \right) \quad (5.10)$$

où :

- $\mathcal{H}(\omega)$ pour $\omega \in \overline{\Omega} \setminus \partial\overline{\Omega}$ a été définie plus tôt comme une factorisation de Cholesky (voir la section 5.2)
- les $(V_k)_{k \in B_N}$ sont des vecteurs gaussiens à valeurs dans \mathbb{C}^d iid selon la loi normale centrée réduite complexe de \mathbb{C}^d

Remarque 5.5.3. $\mathcal{H}(\omega_k)V_k$ est un produit matrice-vecteur

Remarque 5.5.4. V un vecteur aléatoire à valeurs dans \mathbb{C}^d suit la loi normale centrée réduite complexe si les parties réelle et imaginaire $\Re(V)$ et $\Im(V)$ suivent la loi normale centrée réduite de \mathbb{R}^d et si $\Re(V)$ et $\Im(V)$ sont des vecteurs indépendants.

Finalement on décide d'approcher le champ aléatoire X par X_N en les points de simulation $(t_m)_{m \in B_N}$ définis plus tôt (voir la section 5.4). On cherche donc à simuler $(X_N(t_m, \cdot))_{m \in B_N}$. Cette simulation est possible à l'aide d'un algorithme de simulation qui se déduit directement de la formule (5.10) du champ aléatoire X_N .

5.6 Algorithme de simulation

5.6.1 Premier algorithme

Pour modéliser les (V_k) de la section précédente, il suffit de considérer $(U_{k,b,p})_{k \in B_N, b \in \llbracket 0;1 \rrbracket, p \in \llbracket 1;d \rrbracket}$ une famille de $2 \times Nd$ variables iid selon la loi normale centrée réduite de \mathbb{R} et de poser pour $k \in B_N$, V_k vecteur aléatoire à valeurs dans \mathbb{C}^d tel que :

$$\Re(V_k) = (U_{k,0,1}, \dots, U_{k,0,d}) \quad \text{et} \quad \Im(V_k) = (U_{k,1,1}, \dots, U_{k,1,d})$$

On obtient alors l'algorithme de simulation suivant :

- Soit $(u_{k,b,p})_{k \in B_N, b \in \llbracket 0;1 \rrbracket, p \in \llbracket 1;d \rrbracket}$ une réalisation de taille $2 \times Nd$ de la loi centrée réduite.
- On pose pour $k \in B_N$, $v_k \in \mathbb{C}^d$ tel que pour $p \in \llbracket 1;d \rrbracket$, la p -ième coordonnée de v_k vaut $u_{k,0,p} + iu_{k,1,p}$
- Pour $m \in B_N$ on pose :

$$x_N(t_m) = \Re \left(\sum_{k \in B_N} \sqrt{\Delta\omega} e^{i \langle t_m, \omega_k \rangle} \mathcal{H}(\omega_k) v_k \right) \quad (5.11)$$

Les $x_N(t_m)$ constituent une réalisation du champ discret $(X_N(t_m))_{m \in B_N}$.

5.6.2 Reformulation avec la densité spectrale fréquentielle

Il est possible que l'on n'ait pas accès à la densité spectrale de X , S_X , mais à sa densité spectrale fréquentielle S_X^{fr} (voir la sous-section A.3.3 de l'annexe A). Dans ce cas, il est possible de reformuler l'algorithme de simulation en imposant quelques notations :

- pour $k \in B_N$, $f_k = \frac{1}{2\pi} \cdot \omega_k \in \mathbb{R}^n$ (fréquence associée aux ω_k)
- $\overline{D_f} = \frac{1}{2\pi} \overline{\Omega} = [-\frac{\Omega_{c1}}{2\pi}, \frac{\Omega_{c1}}{2\pi}] \times \dots \times [-\frac{\Omega_{cn}}{2\pi}, \frac{\Omega_{cn}}{2\pi}]$ (domaine fréquentiel)
- $\forall j \in \llbracket 1;n \rrbracket$, $\Delta f_j = \frac{1}{2\pi} \Delta \omega_j$

- $\Delta f = \Delta f_1 \dots \Delta f_n \in \mathbb{R}_+^*$
- pour $f \in \overline{D_f} \setminus \partial \overline{D_f}$, $\tilde{H}(f) = (2\pi)^{\frac{n}{2}} \mathcal{H}(2\pi f)$ ($2\pi f \in \overline{\Omega} \setminus \partial \overline{\Omega}$)

Remarquons d'abord que \tilde{H} décrit une décomposition de Cholesky de S_X^{fr} :

$$\forall f \in \overline{D_f} \setminus \partial \overline{D_f}, \tilde{H}(f) \tilde{H}(f)^* = (2\pi)^n \mathcal{H}(2\pi f) \mathcal{H}(2\pi f)^* = (2\pi)^n S_X(2\pi f) = S_X^{fr}(f)$$

Une décomposition de Cholesky de S_X^{fr} permet alors de réécrire les $x_N(t_m)$ introduits dans l'algorithme de simulation (voir la formule (5.11)) à l'aide des notations posées juste avant :

$$\forall m \in B_N, x_N(t_m) = \Re \left(\sum_{k \in B_N} \sqrt{\Delta f} e^{i \langle t_m, 2\pi f_k \rangle_n} \tilde{H}(f_k) v_k \right)$$

Ce résultat indique la similitude des algorithmes de simulation peu importe si on voit le problème en fréquence ou en pulsation.

5.6.3 Autre reformulation de l'approximation

Dans l'annexe A, on a exhibé multiples propriétés de la densité spectrale dont notamment celle-ci : $\forall \omega \in \mathbb{R}^n, S_X(-\omega) = \overline{S_X(\omega)}$. A partir de cette remarque, il est possible de décrire un champ gaussien équivalent à X_N (dans le sens où il suit la même loi que X_N).

Notons A l'ensemble des ω_k

Notons B l'ensemble des ω_k tels que leur première coordonnée soit positive.

Notons $-B$ l'ensemble $\{-b, b \in B\}$

On a supposé que les $(N_j)_{j \in \llbracket 1; n \rrbracket}$ étaient des multiples de 2 (voir l'hypothèse 5.3.1). On a alors que B est l'ensemble des ω_k tels que leur première coordonnée soit strictement positive. De façon plus explicite on a :

$$B = \{\omega_k, \text{ où } k = (k_1, \dots, k_n) \in B_N \text{ et } k_1 \geq \frac{N_1}{2} + 1\} \quad (5.12)$$

Si on note $\tilde{B} = \{k = (k_1, \dots, k_n) \in B_N \text{ tel que } k_1 \geq \frac{N_1}{2} + 1\}$, on montre aussi que :

$$\begin{aligned} A \setminus B &= \{\omega_{(N_1-k_1+1, N_2-k_2+1, \dots, N_n-k_n+1)} \text{ où } k = (k_1, \dots, k_n) \in \tilde{B}\} \\ &= \{-\omega_k \text{ où } k = (k_1, \dots, k_n) \in \tilde{B}\} \\ &= -B \end{aligned}$$

D'où le couple $(B, -B)$ forme une partition de A l'ensemble des ω_k . Il est alors possible de réécrire la formule (5.10) du champ X_N . Si on pose pour $k = (k_1, \dots, k_n) \in B_N$, $j_k = (N_1 - k_1 + 1, N_2 - k_2 + 1, \dots, N_n - k_n + 1)$, alors pour $t \in \mathbb{R}^n$:

$$\begin{aligned}
 X_N(t, \cdot) &= \Re \left(\sum_{k \in B_N} \sqrt{\Delta \omega} e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_k \right) \\
 &= \Re \left(\sum_{k=(k_1, \dots, k_n) \in \tilde{B}} \sqrt{\Delta \omega} (e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_k + e^{i \langle t, \omega_{j_k} \rangle} \mathcal{H}(\omega_{j_k}) V_{j_k}) \right) \\
 &= \Re \left(\sum_{k=(k_1, \dots, k_n) \in \tilde{B}} \sqrt{\Delta \omega} (e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_k + e^{i \langle t, -\omega_k \rangle} \overline{\mathcal{H}(\omega_k)} V_{j_k}) \right) \\
 &= \Re \left(\sum_{k=(k_1, \dots, k_n) \in \tilde{B}} \sqrt{\Delta \omega} (e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_k + \overline{e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_{j_k}}) \right)
 \end{aligned}$$

Remarque 5.6.1. La notation j_k est la définition explicite de la notation j_k introduite à la section 5.5 et utilisée pour la formule (5.7).

On observe que $\overline{V_{j_k}}$ suit la même loi que V_{j_k} et plus généralement que la famille de vecteurs aléatoires composée de ceux de la famille $(V_k)_{k \in \tilde{B}}$ et de ceux de la famille $(\overline{V_{j_k}})_{k \in \tilde{B}}$ forme une famille de variables aléatoires iid qui peut être confondue avec la famille $(V_k)_{k \in B_N}$. Donc si on pose pour $t \in \mathbb{R}^n$:

$$\tilde{X}_N(t, \cdot) = \Re \left(\sum_{k=(k_1, \dots, k_n) \in \tilde{B}} \sqrt{\Delta \omega} (e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_k + \overline{e^{i \langle t, \omega_k \rangle} \mathcal{H}(\omega_k) V_{j_k}}) \right) \quad (5.13)$$

alors ce champ \tilde{X}_N suit la même loi que le champ X_N . Par conséquent, en raisonnant comme à la sous-section 5.6.1, on peut en déduire sans souci un deuxième algorithme pour simuler les $(\tilde{X}_N(t_m))_{m \in B_N}$. L'avantage de ce nouvel algorithme sera de diminuer les calculs. En effet la connaissance des $(\mathcal{H}(\omega_k))_{k \in \tilde{B}}$, qui s'obtiennent par la méthode de Cholesky, suffit. D'où 2 fois moins de calculs (\tilde{B} est de cardinal $N/2$).

5.7 Algorithme de simulation et TFD

Si on reconsidère dans la sous-section 5.6.1, la formule (5.11) des $x_N(t_m)$ qui décrit le premier algorithme de simulation, la formule se veut sous une forme matricielle très condensée. De plus cette formule n'utilise pas la définition des t_m et des ω_k (voir les sections 5.3 et 5.4). Il est donc intéressant de développer la formule des $x_N(t_m)$ pour chaque coordonnée p allant de 1 à d (que l'on va noter $x_{N,p}(t_m)$). Dans cette section, on met en avant la présence

d'une transformée de Fourier discrète (TFD), dont l'algorithme associé se veut particulièrement efficace.

Pour $m \in B_N$ et $k \in B_N$, on obtient en développant le produit scalaire suivant :

$$\begin{aligned} \langle t_m, \omega_k \rangle_n &= \sum_{j=1}^n t_{j,m_j} \omega_{j,k_j} \\ &= \sum_{j=1}^n \left(-\frac{\pi(m_j-1)(N_j-1)}{N_j} + \frac{2\pi(m_j-1)(k_j-1)}{N_j} \right) \end{aligned}$$

Si on utilise cette formule dans la formule des $x_N(t_m)$, on obtient alors pour $m \in B_N$ et $p \in \llbracket 1; d \rrbracket$:

$$x_{N,p}(t_m) = \sqrt{\Delta\omega} \Re \left(y_{N,p}(m) \cdot \exp \left(-i\pi \sum_{j=1}^n \frac{(m_j-1)(N_j-1)}{N_j} \right) \right)$$

où :

$$- y_{N,p}(m) = \sum_{k \in B_N} z_p(k) \cdot \exp \left(2i\pi \sum_{j=1}^n \frac{(m_j-1)(k_j-1)}{N_j} \right)$$

$$- z_p(k) = \sum_{q=1}^d \mathcal{H}_{pq}(\omega_k) \cdot (u_{k,0,q} + i \cdot u_{k,1,q}) \text{ pour } k \in B_N$$

On observe pour $p \in \llbracket 1; d \rrbracket$ que la famille $(y_{N,p}(m))_{m \in B_N}$ n'est rien d'autre que la transformée de Fourier discrète (TFD) de la famille $(z_p(k))_{k \in B_N}$. De façon générale si $n \in \llbracket 1; 3 \rrbracket$, l'algorithme de simulation admet une complexité en temps en $O(dN \log(N) + d^3 N)$ à cause des d TFDs et du calcul des factorisations de Cholesky. Comme d varie en général entre 1 et 3 on peut parler d'une complexité en $O(N \log(N))$. La complexité en mémoire est en $O(d^2 N)$, en raison du stockage des N factorisations de Cholesky.

5.8 Propriétés des approximations X_N

Dans cette section, on répertorie un ensemble de propriétés vérifiées par l'approximation du champ gaussien X , X_N , introduit à la section 5.5 par la formule 5.10.

— X_N est un champ gaussien centré, homogène tel que :

$$\mathbb{E}(\|X_N\|_2^2) = \Delta\omega \cdot \sum_{k \in B_N} \text{tr}(S_X(\omega_k)) \quad (5.14)$$

- Si on note R_{X_N} sa fonction d'autocovariance qui à $s \in \mathbb{R}^n$ associe $\mathbb{E}(X_N(0)X_N(s))^T$, alors :

$$R_{X_N}(s) = \Delta\omega \cdot \Re \left(\sum_{k \in B_N} e^{i\langle s, \omega_k \rangle} S_X(\omega_k) \right) \quad (5.15)$$

On remarque que $\sum_{k \in B_N} e^{i\langle s, \omega_k \rangle} S_X(\omega_k)$ est un réel

- X_N est continue en moyenne quadratique et admet une mesure spectrale M_{X_N} à valeurs dans $M_d(\mathbb{R})$ tel que :

$$M_{X_N} = \Delta\omega \cdot \sum_{k \in B_N} S_X(\omega_k) \delta_{\omega_k} \quad (5.16)$$

où δ_{ω_k} est la mesure de Dirac en ω_k (mesure sur $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$)

- X_N est presque sûrement à trajectoires continues.
- Les trajectoires de X_N sont de période $\tau = (\tau_1, \dots, \tau_n)$ où $\forall j \in \llbracket 1; n \rrbracket$, $\tau_j = \frac{4\pi}{\Delta\omega_j} = 2T_j$ (i.e $\forall j \in \llbracket 1; n \rrbracket$, $\forall t \in \mathbb{R}^n$, $X_N(t_1, \dots, t_j + \tau_j, \dots, t_n) = X_N(t_1, \dots, t_j, \dots, t_n)$)

Considérons maintenant une suite $(N^j = (N_1^j, \dots, N_n^j))_{j \in \mathbb{N}}$ à valeurs dans \mathbb{N}^n . Pour $j \in \mathbb{N}$, on note $N_{inf}^j = \inf_{i \in \llbracket 1; n \rrbracket} N_i^j$. On suppose que $N_{inf}^j \xrightarrow{j \rightarrow \infty} \infty$.

Notons maintenant pour $j \in \mathbb{N}$, X_{N^j} le champ aléatoire défini par la formule (5.10), $M_{X_{N^j}}$ sa mesure spectrale définie par la formule 5.16 et $R_{X_{N^j}}$ la fonction d'autocovariance de X_{N^j} définie par la formule (5.15) où dans les formules la discrétisation (N_1, \dots, N_n) qui définit les ω_k est remplacée par la discrétisation (N_1^j, \dots, N_n^j) .

On obtient alors des propriétés de convergence :

- La suite $(X_{N^j})_{j \in \mathbb{N}}$ converge en loi vers le champ aléatoire X .
- La suite $(R_{N^j})_{j \in \mathbb{N}}$ converge uniformément dans tout compact de \mathbb{R}^n vers R_X la fonction d'autocovariance de X .
- Pour tout $(k, l) \in \llbracket 1; d \rrbracket^2$, la suite de mesures complexes $((M_{N^j})_{k,l})_{j \in \mathbb{N}}$ converge étroitement vers la mesure $(M_X)_{k,l}$.
- Si S_X est de classe C^1 (resp. C^2) sur $\bar{\Omega}$, alors la vitesse des 2 convergences précédentes est en $(N_{inf}^j)^{-1}$ (resp. $(N_{inf}^j)^{-2}$).

5.9 Erreur et convergence numérique

5.9.1 Nouvel estimateur

Dans le cas de la méthode spectrale, on ne choisira pas d'estimer l'erreur L^2 de N réalisations (voir la sous-section 2.2.1 du chapitre 2) afin de valider cette méthode de simulation. On rappelle que l'erreur L^2 permet de voir à quel point par N réalisations, on approche la matrice de covariance Σ du problème. Or même si à partir d'une densité spectrale on peut obtenir par calcul la fonction d'autocovariance d'un processus (et donc la fonction de covariance dont on déduit la matrice Σ), on peut être dans une situation où seule l'information de la densité spectrale est disponible. C'est donc cette dernière que l'on cherchera à estimer à l'aide de L réalisations indépendantes du même processus sur le domaine de simulation. Explicitons l'estimateur évoqué dans la référence [2].

Soit $(x^{(1)}(t_m))_{m \in B_N}, \dots, (x^{(L)}(t_m))_{m \in B_N}$ L réalisations indépendantes du processus X_N (voir la formule (5.10) de la section 5.5) en les points de simulation $(t_m)_{m \in B_N}$ (voir la section 5.4). Pour valider nos simulations, on décide d'utiliser plus précisément un estimateur de la densité spectrale fréquentielle S_X^{fr} en les points $(f_k)_{k \in B_N} = (\frac{1}{2\pi}\omega_k)_{k \in B_N}$ (voir formule (5.2) de la section 5.3) que l'on qualifiera de points fréquents associés aux points de simulation. On définit alors pour $k = (k_1, \dots, k_n) \in B_N$ l'estimateur de $S_X^{fr}(f_k)$:

$$\hat{S}(f_k) = \frac{1}{L} \sum_{l=1}^L \hat{Z}^{(l)}(k) \hat{Z}^{(l)}(k)^* \in M_d(\mathbb{C}) \quad (5.17)$$

où :

— pour $l \in \llbracket 1; L \rrbracket$, $\hat{Z}^{(l)}(k) = (\hat{Z}_1^{(l)}(k), \dots, \hat{Z}_d^{(l)}(k))^T$

— pour $l \in \llbracket 1; L \rrbracket$, pour $p \in \llbracket 1; d \rrbracket$:

$$\hat{Z}_p^{(l)}(k) = \sum_{m \in B_N} Z_p^{(l)}(m) \cdot \exp\left(-2i\pi \sum_{j=1}^n \frac{(m_j - 1)(k_j - 1)}{N_j}\right)$$

— pour $l \in \llbracket 1; L \rrbracket$, $m \in B_N$ et $p \in \llbracket 1; d \rrbracket$:

$$Z_p^{(l)}(m) = \Delta t \cdot W_{\bar{T}}(t_m) \cdot x_p^{(l)}(t_m) \cdot \exp\left(i\pi \sum_{j=1}^n \frac{(m_j - 1)(N_j - 1)}{N_j}\right)$$

— pour $s \in \mathbb{R}^n$, $W_{\bar{T}}(s) = W_{T_1}(s_1) \cdots W_{T_n}(s_n)$ (on parle de fenêtre temporelle de dimension n sur $\bar{T} = [0, T_1] \times \cdots \times [0, T_n]$)

— pour $s \in \mathbb{R}$ et $j \in \llbracket 1; n \rrbracket$, $W_{T_j}(s) = \frac{1}{\sqrt{T_j}} W(\frac{s}{T_j})$ où W est une fenêtre temporelle de dimension 1 standard.

Définition 5.9.1. Une fenêtre temporelle d'énergie normalisée standard ou juste fenêtre temporelle de dimension 1 standard est une fonction $W : \mathbb{R} \rightarrow \mathbb{R}$ mesurable telle que :

- $\text{supp}(W) \subset [0, 1]$ (support de W inclus dans $[0, 1]$)
- $W(1 - t) = W(t)$ pour $t \in \mathbb{R}$
- $\int_{\mathbb{R}} W(t)^2 dt = \int_0^1 W(t)^2 dt = 1$
- si on note \hat{W} la transformée de Fourier de W et que l'on note pour $T \in \mathbb{R}_+^*$ et $\omega \in \mathbb{R}$, $G_T(\omega) = \frac{T}{2\pi} |\hat{W}(T\omega)|^2$, alors quand T tend vers l'infini, G_T converge en distribution vers la masse de Dirac en 0.

Remarque 5.9.1. $\hat{W}(\omega) = \int_{\mathbb{R}} e^{-i\omega t} W(t) dt$

Remarque 5.9.2. Pour $T \in \mathbb{R}_+^*$, $W_T : t \rightarrow \frac{1}{\sqrt{T}} W(\frac{t}{T})$ est appelé fenêtre temporelle et vérifie les propriétés suivantes :

- $\text{supp}(W_T) \subset [0, T]$
- $W_T(T - t) = W_T(t)$ pour $t \in \mathbb{R}$
- $\int_{\mathbb{R}} W_T(t)^2 dt = \int_0^T W_T(t)^2 dt = 1$
- $\frac{1}{2\pi} |\hat{W}_T(\omega)|^2 = G_T(\omega)$ pour $\omega \in \mathbb{R}$ (on dit alors que G_T est la fenêtre spectrale associée à W_T)

Dans la référence [12], plus de détails sur les fenêtres temporelles sont évoqués et leur rôle est d'essayer d'améliorer les estimations d'une densité spectrale. On peut citer comme exemple de fenêtre temporelle standard, la fenêtre de Hamming que l'on utilisera pour nos tests numériques et qui est définie de la façon suivante pour $t \in \mathbb{R}$:

$$W(t) = 1.5863 (0.54 - 0.46 \cos(2\pi t)) \mathbb{1}_{[0,1]}(t) \quad (5.18)$$

Dans les références [2] et [12], la pertinence (dans le cas des champs gaussiens surtout) d'utiliser un tel estimateur pour la densité spectrale S_X^{fr} en augmentant le nombre de réalisations L y est un peu plus développé mais on n'entrera pas dans les détails.

Finalement on quantifie l'erreur de simulation en définissant l'erreur absolue ϵ suivante :

$$\epsilon = \max_{k \in B_N} \|\hat{S}(f_k) - S_X^{fr}(f_k)\|_F \quad (5.19)$$

5.9.2 Résultats numériques

Dans l'annexe B, dans la section B.4, plusieurs densités spectrales y sont évoquées et ont une notation : (S1) et sa fonction d'autocovariance associée (R1), (S2) et sa fonction d'autocovariance associée (R2), etc... On utilisera cette notation afin de les désigner par la suite. Par ailleurs on maintient

toujours les notations utilisées jusqu'à maintenant pour décrire la méthode spectrale : $N_1, \dots, N_n, f_{max,1}, \dots, f_{max,n}$ et ainsi de suite. Pour nos tests, $n \in \llbracket 1; 3 \rrbracket$ et bien qu'il ne soit pas nécessaire d'associer un maillage aux points de simulation, en raison de la répartition régulière dans l'espace des points de simulation, on peut y associer tout de même un maillage triangulaire dont les nœuds sont les points de simulations et dont les n -simplexes sont définis de la même façon que pour les maillages évoqués à la section 2.3 du chapitre 2.

Les sous-sous-sections suivantes décrivent les différents tests numériques effectués. Ces derniers sont contenus dans le fichier spectral.py (voir la section B.4 de l'annexe B). Pour tester l'estimateur de densité spectrale, il a fallu choisir le nombre de réalisations indépendantes L du processus X_N et on l'a fait varier entre 100 et 2000. L'erreur de simulation sera l'erreur absolue ϵ décrite par la formule (5.19). Pour le choix de la fenêtre temporelle standard, on opte pour la fenêtre de Hamming.

5.9.2.1 Dimension n=1

En dimension 1, on a choisi la densité spectrale (fréquentielle) (S1) où le paramètre A a été fixé à 5.0. Comme la densité (S1) est à support dans $[-A, A]$, il est pertinent de choisir $f_{max,1} = A$. Le nombre de points de simulation $N = N_1$ varie entre 10 et 10000. La connaissance de N et $f_{max,1}$ permet de déduire les N points de simulation $(\frac{m-1}{2f_{max,1}})_{m \in \llbracket 1; N \rrbracket}$. Le point de simulation de valeur maximale est donc le point $t_N = \frac{N-1}{2f_{max,1}}$, qui est le temps effectif de simulation en contraste avec le temps de simulation $T_1 = \frac{N}{2f_{max,1}}$. Aux points de simulation, on associe les points fréquents :

$$\left((2i-1-N) \frac{f_{max,1}}{N_1} \right)_{i \in \llbracket 1; N \rrbracket}$$

5.9.2.2 Dimension n=2

En dimension 2, on a choisi la densité spectrale (S4) où le paramètre σ vaut 1 et les paramètres b_1 et b_2 ont été fixés respectivement à 1 et 4. On a fixé $f_{max,1} = \frac{20}{2\pi}$ et $f_{max,2} = \frac{5}{2\pi}$. Le nombre de points de simulation $N = N_1 \times N_2$ varie entre 10 et 10000 et $N_1 = N_2$. La connaissance de N , $f_{max,1}$ et $f_{max,2}$ permet de déduire les N points de simulation :

$$\left(\left(\frac{i-1}{2f_{max,1}}, \frac{j-1}{2f_{max,2}} \right) \right)_{(i,j) \in \llbracket 1; N_1 \rrbracket \times \llbracket 1; N_2 \rrbracket}$$

auxquels on associe les points fréquents :

$$\left(\left((2i-1-N_1) \frac{f_{max,1}}{N_1}, (2j-1-N_1) \frac{f_{max,2}}{N_2} \right) \right)_{(i,j) \in \llbracket 1; N_1 \rrbracket \times \llbracket 1; N_2 \rrbracket}$$

5.9.2.3 Dimension n=3

En dimension 3, on a choisi la densité spectrale (S3) où le paramètre a a été fixé à 1.0. Les fréquences maximales $f_{max,1}, f_{max,2}, f_{max,3}$ ont été fixées toutes trois à 10. Le nombre de points de simulation $N = N_1 \times N_2 \times N_3$ varie entre 10 et 10000 et $N_1 = N_2 = N_3$. La connaissance de N , $f_{max,1}$, $f_{max,2}$ et $f_{max,3}$ permet de déduire les N points de simulation :

$$\left(\left(\frac{i-1}{2f_{max,1}}, \frac{j-1}{2f_{max,2}}, \frac{k-1}{2f_{max,3}} \right) \right)_{(i,j,k) \in \llbracket 1; N_1 \rrbracket \times \llbracket 1; N_2 \rrbracket \times \llbracket 1; N_3 \rrbracket}$$

auxquels on associe les points fréquentiels :

$$\left(\left((2i-1-N_1) \frac{f_{max,1}}{N_1}, (2j-1-N_1) \frac{f_{max,2}}{N_2}, (2k-1-N_1) \frac{f_{max,3}}{N_3} \right) \right)_{(i,j,k) \in \llbracket 1; N_1 \rrbracket \times \llbracket 1; N_2 \rrbracket \times \llbracket 1; N_3 \rrbracket}$$

5.9.2.4 Benchmarks

Dimension	Nb de nœuds N	Temps moyen d'une réalisation (en seconde)
1	10	0.00024s
1	100	0.00196s
1	1000	0.01924s
1	10000	0.19921s
2	16	0.00060s
2	100	0.00271s
2	1024	0.02546s
2	10000	0.24736s
3	8	0.00040s
3	64	0.00200s
3	1000	0.02714s
3	10648	0.27015s

Dimension	Nb de nœuds N	Nb de réalisations L	erreur ϵ
1	100	100	0.02323
1	100	500	0.00803
1	100	1000	0.00968
1	100	2000	0.00485
1	500	100	0.02758
1	500	500	0.01412
1	500	1000	0.00815
1	500	2000	0.00835
1	1000	100	0.03043
1	1000	500	0.01536
1	1000	1000	0.01083
1	1000	2000	0.00751
2	100	100	0.56987
2	100	500	0.39173
2	100	1000	0.42713
2	100	2000	0.45083
2	484	100	5.07162
2	484	500	2.54857
2	484	1000	2.94699
2	484	2000	3.13654
2	1024	100	2.01205
2	1024	500	1.83634
2	1024	1000	0.93156
2	1024	2000	1.05494
3	64	100	$1.80454 \cdot 10^{-5}$
3	64	500	$1.66974 \cdot 10^{-5}$
3	64	1000	$1.62478 \cdot 10^{-5}$
3	64	2000	$1.52534 \cdot 10^{-5}$
3	512	100	0.00027
3	512	500	0.00024
3	512	1000	0.00021
3	512	2000	0.00022
3	1000	100	0.00065
3	1000	500	0.00062
3	1000	1000	0.00054
3	1000	2000	0.00057

Remarque 5.9.3. Dans le tableau de l'erreur ϵ , l'erreur ne décroît pas forcément bien vers 0 et certains niveaux d'erreur sont étonnamment élevés (dimension 2). On soupçonne un bug qui est toujours en cours d'investigation au moment de la rédaction du rapport.

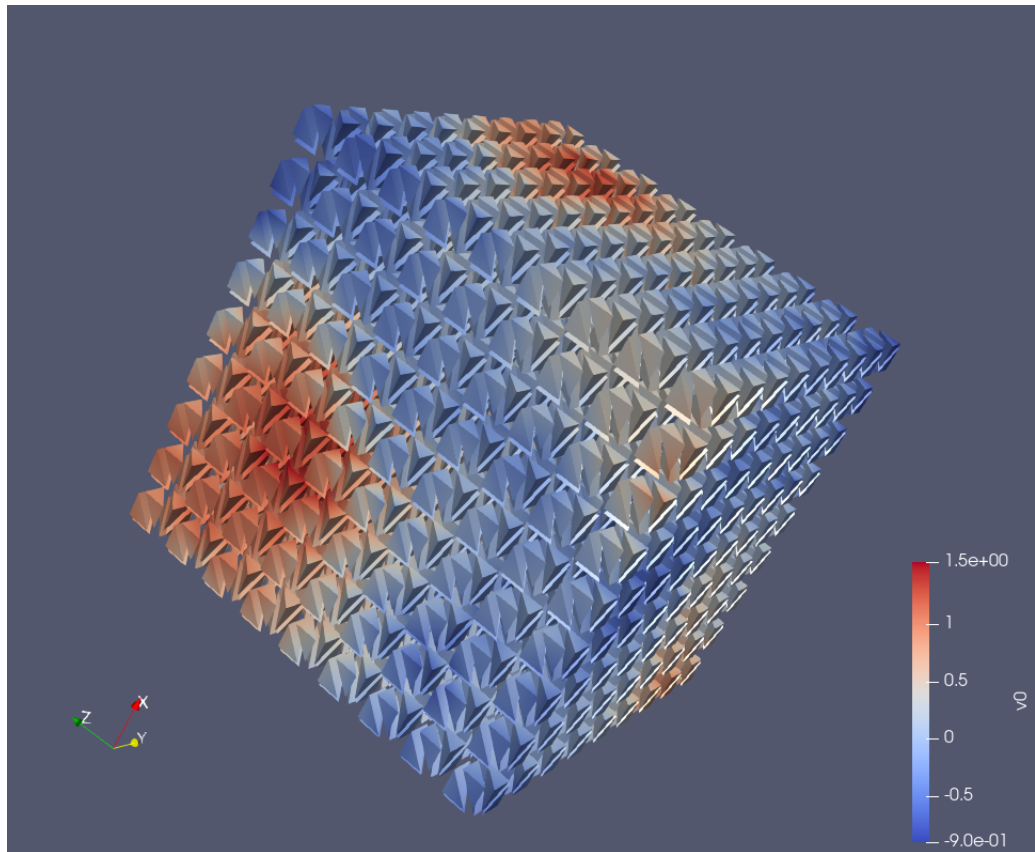


FIGURE 5.3 – Réalisation en dimension 3 où $N = 1000$

Chapitre 6

Simulation par \mathbb{P}_1 -interpolation

Supposons pouvoir simuler un champ gaussien X sur les nœuds d'un maillage M décrivant la fermeture d'un ouvert borné D . Par une interpolation \mathbb{P}_1 , il est possible à partir de toutes les réalisations du processus X connu en les nœuds de M , d'obtenir un nouveau processus \tilde{X} sur \bar{D} . On aimerait que le processus \tilde{X} devienne une bonne approximation du processus X (pour un maillage M suffisamment raffiné) en les nœuds d'un nouveau maillage M' inclus dans \bar{D} .

L'intérêt pratique de la \mathbb{P}_1 -interpolation est que certaines méthodes de simulation sont très efficaces sur des maillages M comme les maillages semblables à ceux de la méthode spectrale du chapitre précédent ou comme les maillages ne présentant pas une concentration de nœuds (efficacité de la compression en \mathcal{H} -matrice). D'où l'idée de transférer par \mathbb{P}_1 -interpolation sur le maillage d'intérêt M' , les résultats d'une simulation efficace sur M .

6.1 Cadre théorique

Soit $(\Omega, \mathcal{F}, \mathbb{P})$ un espace probabilisé

Soit n, d deux entiers naturels non nuls et D un ouvert borné de \mathbb{R}^n

On note λ^n la mesure de Lebesgue sur D

On admet aussi que D peut être décrit par un maillage triangulaire.

Définition 6.1.1. On dit que le champ aléatoire $X : D \times \Omega \rightarrow \mathbb{R}^d$ est un champ aléatoire mesurable si X est une fonction mesurable, où $D \times \Omega$ a été muni de la tribu issue de la complétion de la mesure-produit $\lambda^n \otimes \mathbb{P}$

En se référant à [11], on obtient que tout champ aléatoire X d'ordre 2 dont les éléments diagonaux de la fonction de covariance sont des fonctions continues admet une version Y qui est mesurable. Par conséquent, pour qu'un champ gaussien puisse être considéré comme un champ aléatoire mesurable, il suffit de supposer que les éléments diagonaux de sa fonction de covariance

soient des fonctions continues.

Pour étudier les champs aléatoires mesurables, on introduit également l'espace de Hilbert $(L^2_{\mathbb{R}^d}(D \times \Omega, \lambda^n \otimes \mathbb{P}), \|\cdot\|)$, l'ensemble des fonctions ϕ de $D \times \Omega$ dans \mathbb{R}^d telles que :

- ϕ est $(\mathcal{B}(D) \otimes \mathcal{F}, \mathcal{B}(\mathbb{R}^d))$ -mesurable
- la quantité $\|\phi\|^2 = \mathbb{E}(\int_D \|\phi(t, \cdot)\|_2^2 d\lambda^n(t))$ est finie

Ainsi, un champ aléatoire mesurable X est dans $L^2_{\mathbb{R}^d}(D \times \Omega, \lambda^n \otimes \mathbb{P})$ si et seulement si $\|X\|$ est une quantité finie.

6.2 Approximation d'une fonction par méthode des éléments finis \mathbb{P}_1

L'approximation \mathbb{P}_1 -Lagrange est une manière très classique en analyse numérique d'approcher la solution d'une EDP (équation aux dérivées partielles). Par conséquent, tout lecteur qui souhaitera des détails sur ce procédé pourra se référer au livre de Grégoire Allaire [1]. De ce livre, on utilise notamment les définitions de maillage triangulaire ou triangulation (page 175) et de suite de maillages réguliers (page 188).

Rappelons rapidement en quoi consiste l'approximation d'une fonction par la méthode des éléments finis. On considère un maillage triangulaire M décrivant la fermeture de D (D peut être supposé connexe). Si on note $(n_i)_{i \in \llbracket 1, m \rrbracket}$ les nœuds de M , on dispose d'un unique m -uplet (ψ_1, \dots, ψ_m) de fonctions continues de \bar{D} dans \mathbb{R} et polynomiales de degré 1 sur chaque n -simplexe du maillage M et tel que :

$$\forall (i, j) \in \llbracket 1, m \rrbracket^2, \psi_i(n_j) = \delta_{i,j}$$

Remarque 6.2.1. On appelle les ψ_i , les fonctions de base du maillage M

On définit alors pour $f : \bar{D} \rightarrow \mathbb{R}^d$ une fonction continue sur D , $R_{M,d}(f)$, l'interpolée \mathbb{P}_1 de f comme la fonction telle que :

$$\forall t \in \bar{D}, R_{M,d}(f)(t) = \sum_{i=1}^m \psi_i(t) \cdot f(n_i) \quad (6.1)$$

Remarque 6.2.2. $R_{M,d}(f)$ prend les mêmes valeurs que f en les nœuds du maillage

L'opérateur d'interpolation $R_{M,d}$ décrit l'approximation par interpolation \mathbb{P}_1 -Lagrange des fonctions continues sur \bar{D} . Dans le livre de Grégoire Allaire,

tout est présenté dans le cas où $d = 1$, notamment les espaces de Sobolev qui seront utiles pour faire des estimations. Cependant, il n'est guère difficile de généraliser la définition des espaces de Sobolev dans le cas $d > 1$ comme il va suivre.

6.3 Espaces de Sobolev et interpolation

On suppose que le lecteur est familier avec les espaces de Sobolev notés $W^{k,p}(O)$ ou $H^k(O)$, où O est un ouvert. On peut ainsi présenter une légère généralisation des espaces de Sobolev où les (classes de) fonctions sont à valeurs dans \mathbb{R}^d , où d est éventuellement strictement supérieur à 1.

Définition 6.3.1. Pour O un ouvert de \mathbb{R}^n , pour $k \in \mathbb{N}^*$ et $p \in [1, \infty]$:

$$W_{\mathbb{R}^d}^{k,p}(O) = \{f = (f_1, \dots, f_d)^T : O \rightarrow \mathbb{R}^d / \forall i \in \llbracket 1, d \rrbracket, f_i \in W^{k,p}(O)\}$$

On se servira par la suite surtout de cet ensemble :

$$H_{\mathbb{R}^d}^k(O) = W_{\mathbb{R}^d}^{k,2}(O)$$

Propriété 6.3.1. Pour $k \in \mathbb{N}^*$, l'espace $H_{\mathbb{R}^d}^k(O)$ est un espace de Hilbert s'il est muni de la norme $\|\cdot\|_{H_{\mathbb{R}^d}^k(O)}$ telle que pour $f \in H_{\mathbb{R}^d}^k(O)$:

$$\|f\|_{H_{\mathbb{R}^d}^k(O)}^2 = \sum_{i=1}^d \|f_i\|_{H^k(O)}^2$$

Avec ces définitions, notons les 2 propriétés suivantes :

Propriété 6.3.2. Pour tout maillage triangulaire M de \bar{D} et toute fonction continue $f : \bar{D} \rightarrow \mathbb{R}^d$, $R_{M,d}(f) \in H_{\mathbb{R}^d}^1(D)$

Propriété 6.3.3 (Injection de Sobolev). Soit $k \in \mathbb{N}$ tel que $k + 1 > n/2$. $\forall w \in H_{\mathbb{R}^d}^{k+1}(D)$, w est la restriction sur D d'une fonction continue sur \bar{D} .

On est maintenant apte à introduire un théorème majeur pour la suite ainsi que son corollaire. Considérons $(\mathcal{T}_h)_{h \in \mathbb{R}_+^*}$ une suite de maillages (triangulaires) réguliers de \bar{D} . On rappelle que le paramètre h désigne le maximum des diamètres des n -simplexes qui composent le maillage \mathcal{T}_h . Pour $h \in \mathbb{R}_+^*$, on note $R_{h,d}$ l'opérateur d'interpolation $R_{\mathcal{T}_h,d}$.

Théorème 6.3.1. Soit $k \in \mathbb{N}$ tel que $k + 1 > n/2$. Il existe $C \in \mathbb{R}_+^*$ tel que :

$$\forall h \in \mathbb{R}_+^*, \forall v \in H^{k+1}(D), \|v - R_{h,1}(v)\|_{H^1(D)} \leq Ch^k \|v\|_{H^{k+1}(D)}$$

Corollaire 6.3.1. Soit $k \in \mathbb{N}$ tel que $k + 1 > n/2$. Il existe $C \in \mathbb{R}_+^*$ tel que :

$$\forall h \in \mathbb{R}_+^*, \forall v \in H_{\mathbb{R}^d}^{k+1}(D), \|v - R_{h,d}(v)\|_{H_{\mathbb{R}^d}^1(D)} \leq Ch^k \|v\|_{H_{\mathbb{R}^d}^{k+1}(D)}$$

6.4 Espaces de Sobolev et champs aléatoires

Lorsque l'on étudie les réalisations d'un champ aléatoire, on peut s'interroger sur la régularité des trajectoires (continuité, différentiabilité). Dans cette section, on exhibe des conditions pour qu'un champ aléatoire mesurable (voir la définition 6.1.1) puisse admettre presque sûrement que des trajectoires dans un espace de Sobolev. Tout ce qui suit est fondé en grande partie sur le livre de Hristopulos [6], l'article de Scheuerer [11] et le chapitre 4 de "*The Theory of Stochastic Processes I*" [5].

6.4.1 Sur la différentiabilité en moyenne quadratique d'un champ aléatoire

À la page 183 dans [6], pour $X : D \times \Omega \rightarrow \mathbb{R}$ un champ aléatoire d'ordre 2 et $i \in \llbracket 1; n \rrbracket$, on définit, si elle existe, la dérivée partielle de X en moyenne quadratique par rapport à la $i^{\text{ème}}$ variable que l'on note $\frac{\partial}{\partial x_i} X$ ou $\frac{\partial X}{\partial x_i}$. On remarque que sa définition ne revient qu'à dire que l'application $\gamma : D \rightarrow L^2(\Omega, \mathbb{P})$ qui à $t \in D$ associe $\gamma(t) = X(t, \cdot)$ admet une dérivée partielle par rapport à la $i^{\text{ème}}$ variable sur D .

On peut alors définir, si elles existent, les dérivées partielles en moyenne quadratique sur D d'ordre supérieur à 1 de X , $\frac{\partial^k}{\partial x_{\alpha_1} \dots \partial x_{\alpha_k}} X$ où $\alpha_1, \dots, \alpha_k$ sont k entiers entre 1 et n et où la notation indique qu'on a dérivé par rapport à la $\alpha_k^{\text{ème}}$ variable (en moyenne quadratique), puis par rapport à la $\alpha_{k-1}^{\text{ème}}$ variable, ainsi de suite et à la fin par rapport à la $\alpha_1^{\text{ème}}$ variable. On dira alors que $\frac{\partial^k}{\partial x_{\alpha_1} \dots \partial x_{\alpha_k}} X$ est une dérivée partielle (en moyenne quadratique) de X d'ordre k .

On s'apprête à mettre en avant une condition suffisante pour qu'un champ aléatoire d'ordre 2 et centré admette des dérivées partielles en moyenne quadratique. Mais il faut d'abord introduire le concept de dérivée généralisée mixte. Considérons $X : D \times \Omega \rightarrow \mathbb{R}$ un champ aléatoire d'ordre 2, centré et dont on note la fonction de covariance $C : D \times D \rightarrow \mathbb{R}$

Définition 6.4.1. Soit $i \in \llbracket 1; n \rrbracket$ et $(s, t) \in D \times D$. On note e_i le $i^{\text{ème}}$ vecteur de la base canonique de \mathbb{R}^n .

On dit alors que C admet une dérivée mixte généralisée en (s, t) par rapport à la $i^{\text{ème}}$ variable si

$$\lim_{(h,k) \rightarrow (0,0)} \frac{C(s + h \cdot e_i, t + k \cdot e_i) - C(s, t + k \cdot e_i) - C(s + h \cdot e_i, t) + C(s, t)}{hk} \quad (6.2)$$

existe. Si tel est le cas, on notera cette limite $(D^{i,i}C)(s, t)$.

Remarque 6.4.1. On peut voir $D^{i,i}$ comme un opérateur ayant pour ensemble de départ et d'arrivée l'ensemble des fonctions allant de $D \times D$ dans \mathbb{R}

Théorème 6.4.1. Pour $i \in \llbracket 1; n \rrbracket$, X admet une dérivée partielle en moyenne quadratique par rapport à la $i^{\text{ème}}$ variable si et seulement $\forall t \in D$, $(D^{i,i}C)(t, t)$ existe et est fini.

Si tel est le cas, alors $\forall (s, t) \in D \times D$, $(D^{i,i}C)(s, t)$ existe et est finie. De plus :

$$\mathbb{E} \left(\frac{\partial X}{\partial x_i}(s) \frac{\partial X}{\partial x_i}(t) \right) = (D^{i,i}C)(s, t) \quad (6.3)$$

Remarque 6.4.2. Ce théorème explicite donc la fonction de covariance de $\frac{\partial X}{\partial x_i}$.

Définition 6.4.2. Soit $\alpha_1, \dots, \alpha_k$ k entiers entre 1 et n .

- On définit $D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}$ comme l'opérateur $D^{\alpha_1, \alpha_1} \circ \dots \circ D^{\alpha_k, \alpha_k}$.
- Soit $\kappa \in \mathbb{N}^n$ tel que $\forall i \in \llbracket 1; n \rrbracket$, κ_i indique le nombre d'éléments parmi les k entiers $\alpha_1, \dots, \alpha_k$ qui prennent la valeur i .

Alors si pour tout permutation de $\llbracket 1; k \rrbracket$, $D^{(\alpha_{\sigma(1)}, \alpha_{\sigma(1)}), \dots, (\alpha_{\sigma(k)}, \alpha_{\sigma(k)})}C$ est bien défini sur $D \times D$ et que $D^{(\alpha_{\sigma(1)}, \alpha_{\sigma(1)}), \dots, (\alpha_{\sigma(k)}, \alpha_{\sigma(k)})}C = D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}C$, alors on définit $D^\kappa C$ comme étant la fonction $D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}C$.

Corollaire 6.4.1. Soit $\alpha_1, \dots, \alpha_k$ k entiers entre 1 et n , $(s, t) \in D \times D$.

La dérivée partielle en moyenne quadratique $\frac{\partial^k}{\partial x_{\alpha_1} \dots \partial x_{\alpha_k}} X$ existe si et seulement si $\forall t \in D$, la quantité $(D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}C)(t, t)$ est bien définie et finie.

Si tel est le cas, alors $\forall (s, t) \in D \times D$, $(D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}C)(s, t)$ est défini et fini. De plus :

$$\mathbb{E} \left(\frac{\partial^k X}{\partial x_{\alpha_1} \dots \partial x_{\alpha_n}}(s) \frac{\partial^k X}{\partial x_{\alpha_1} \dots \partial x_{\alpha_n}}(t) \right) = (D^{(\alpha_1, \alpha_1), \dots, (\alpha_k, \alpha_k)}C)(s, t) \quad (6.4)$$

Propriété 6.4.1. Soit $(s, t) \in D \times D$ et $m \in \mathbb{N}$.

Si C est une fonction de classe \mathcal{C}^{2m} sur $D \times D$ alors $\forall \kappa \in \mathbb{N}^d$ tel que $|\kappa| \leq m$, $D^\kappa C$ est bien définie sur $D \times D$ et $\forall (s, t) \in D \times D$:

$$(D^\kappa C)(s, t) = \frac{\partial^{2|\kappa|}}{\partial s_1^{\kappa_1} \dots \partial s_n^{\kappa_n} \partial t_1^{\kappa_1} \dots \partial t_n^{\kappa_n}}(s, t) \quad (6.5)$$

6.4.2 Résultat de régularité des trajectoires

On dédie cette sous-section à un autre résultat important qui permet de déduire une estimation théorique en lien avec la simulation d'un processus gaussien par méthode de \mathbb{P}_1 -interpolation présentée à la section suivante. Ce résultat est tiré de l'article de Scheuerer [11]. À partir de la preuve évoquée dans l'article, il est possible de raffiner un peu plus le propos, ce qui s'avèrera nécessaire pour justifier des propriétés mathématiques de la future méthode.

Théorème 6.4.2. Soit O un ouvert de \mathbb{R}^n , $X : O \times \Omega \rightarrow \mathbb{R}$ un champ aléatoire mesurable dont on note $C : O \times O \rightarrow \mathbb{R}$ la fonction de covariance. Soit $k \in \mathbb{N}^*$. Alors si pour tout $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k$, $D^\alpha C$ existe sur $O \times O$ et est continue sur la diagonale de $O \times O$, alors les trajectoires de X sont presque sûrement dans $H_{loc}^k(O)$ et pour tout $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k$, $\mathbb{P}(\omega)$ -presque sûrement, la dérivée en moyenne quadratique $\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X(., \omega)$ coïncide presque partout (pour la mesure de Lebesgue dans O) avec la dérivée faible d'indice α de $X(., \omega)$, $\partial^\alpha X(., \omega)$.

Si de plus pour tout $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k$:

$$\int_O D^\alpha C(t, t) dt < +\infty \quad (6.6)$$

Alors on a même que les trajectoires de X sont presque sûrement dans $H^k(O)$ (ce qui revient à dire que pour tout $\alpha \in \mathbb{N}^n$, $\mathbb{P}(\omega)$ -presque sûrement $\partial^\alpha X(., \omega)$ est dans $L^2(O)$).

Corollaire 6.4.2. Soit O un ouvert de \mathbb{R}^n , $X = (X_1, \dots, X_d) : O \times \Omega \rightarrow \mathbb{R}^d$ un champ aléatoire mesurable dont on note $C = (C_{i,j}) : O \times O \rightarrow M_d(\mathbb{R})$ la fonction de covariance. Soit $k \in \mathbb{N}^*$.

Si pour tout $i \in \llbracket 1; d \rrbracket$, $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k$, $D^\alpha C_{i,i}$ existe sur $O \times O$, est continue sur la diagonale de $O \times O$, et que

$$\int_O D^\alpha C_{i,i}(t, t) dt < +\infty \quad (6.7)$$

alors les trajectoires de X sont presque sûrement dans $H_{\mathbb{R}^d}^k(O)$ et pour $i \in \llbracket 1; d \rrbracket$ et $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k$, $\mathbb{P}(\omega)$ -presque sûrement, la dérivée en moyenne quadratique $\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X_i(., \omega)$ coïncide dans $L^2(O)$ avec la dérivée faible d'indice α de $X_i(., \omega)$, $\partial^\alpha X_i(., \omega)$.

Avec ce théorème, nous avons tous les outils nécessaires pour justifier les propriétés théoriques de la méthode de simulation par \mathbb{P}_1 -interpolation.

6.5 Méthode de \mathbb{P}_1 -interpolation

6.5.1 Description de la méthode

Considérons $X : \bar{D} \times \Omega \rightarrow \mathbb{R}^d$ un champ aléatoire mesurable gaussien centré dont on note $C : \bar{D} \times \bar{D} \rightarrow M_d(\mathbb{R})$ la fonction de covariance. Soit M un maillage triangulaire de \bar{D} dont on note $(n_i)_{i \in \llbracket 1; m \rrbracket}$ les m nœuds du maillage M . On note $(\psi_i)_{i \in \llbracket 1; m \rrbracket}$ les fonctions de base du maillage M .

On définit alors l'interpolée \mathbb{P}_1 de X (par le maillage M), le processus $\mathcal{R}_{M,d}X$ définit de la fonction suivante :

$$\forall (t, \omega) \in \bar{D} \times \Omega, (\mathcal{R}_{M,d}X)(t, \omega) = \sum_{i=1}^m \psi_i(t) \cdot X(n_i, \omega) \quad (6.8)$$

On appellera l'opérateur $\mathcal{R}_{M,d}$, l'opérateur d'interpolation \mathbb{P}_1 (qu'on applique ici à des champs aléatoires à valeurs dans \mathbb{R}^d).

Remarque 6.5.1. Comme X est un champ gaussien, il s'avère que $\mathcal{R}_{M,d}X$ est aussi un champ gaussien.

À partir de ce nouveau processus, si on dispose de M' un maillage qui décrit la fermeture d'un ouvert inclus dans D , on désire étudier le processus $\mathcal{R}_{M,d}X$ en les nœuds du maillage M' en se demandant s'il approche bien le processus X en les points du maillage M' . La prochaine sous-section donne un début d'élément de réponse en exhibant une forme de convergence mais cette convergence ne dit aucunement qu'en raffinant le maillage M alors l'interpolée converge en loi en les nœuds du maillage M' vers le processus X .

Par ailleurs, une telle méthode suppose que l'on sait simuler X en les nœuds de M . Donc la complexité en temps et en mémoire de l'algorithme associé à la méthode dépend du choix de simulation de X en les nœuds de M . Néanmoins une fois qu'on connaît en les nœuds $(n_i)_{i \in \llbracket 1; m \rrbracket}$ les valeurs $X(n_i, \omega)$, la \mathbb{P}_1 -interpolation qui donne en les nœuds $(\tilde{n}_i)_{i \in \llbracket 1; \tilde{m} \rrbracket}$ de M' les $(\mathcal{R}_{M,d}X)(\tilde{n}_i, \omega)$, admet une complexité algorithmique indépendante du choix de simulation. Sur OpenTURNS, l'implémentation des maillages de dimension n allant de 1 à 3 à l'aide d'une structure hiérarchique de données de type k-d tree permet d'accélérer la \mathbb{P}_1 -interpolation. Sans entrer dans les détails, si on suppose que $d \leq 3$ et que tous les maillages M et M' étudiés admettent une borne uniforme sur le nombre de n -simplexes partageant un même nœud, alors la complexité temporelle de la \mathbb{P}_1 -interpolation est en $O(\tilde{m} \cdot \log(m))$. Sous ces mêmes hypothèses, on obtient une complexité mémoire en $O(m + \tilde{m})$.

6.5.2 Estimation d'erreur

Le théorème suivant donne un cadre pertinent quant à l'utilisation de la méthode de \mathbb{P}_1 -interpolation. Considérons d'abord $(\mathcal{T}_h)_{h \in \mathbb{R}_+^*}$ une suite de maillages réguliers de \bar{D} dont on note $\mathcal{R}_{h,d}$ leur opérateur d'interpolation \mathbb{P}_1 de champs aléatoires à valeurs dans \mathbb{R}^d défini par la formule (6.8) et $R_{h,d}$ leur opérateur d'interpolation \mathbb{P}_1 de fonctions continues de \bar{D} dans \mathbb{R}^d défini par la formule (6.1) pour le maillage \mathcal{T}_h .

Théorème 6.5.1. Soit $k \in \mathbb{N}$ tel que $k + 1 > n/2$. Si pour tout $i \in \llbracket 1; d \rrbracket$, $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k + 1$, $D^\alpha C_{i,i}$ existe sur $D \times D$, est continue sur la diagonale de $D \times D$, et tel que

$$\int_D D^\alpha C_{i,i}(t, t) dt < +\infty \quad (6.9)$$

alors X admet presque sûrement des trajectoires dans $H_{\mathbb{R}^d}^{k+1}(D)$ (donc des trajectoires continues sur \bar{D}) et il existe une constante $K \in \mathbb{R}_+$ tel que $\forall h \in \mathbb{R}_+^*$:

$$\|X - \mathcal{R}_{h,d}X\| \leq K \cdot h^k \quad (6.10)$$

Remarque 6.5.2. $\|\cdot\|$ est la norme définie à la fin de la section 6.1.

Remarque 6.5.3. Ce résultat ne nécessite pas que X soit un champ gaussien. Le résultat reste donc valide sans cette hypothèse.

Démonstration. En utilisant le corollaire 6.4.2, on obtient que les trajectoires de X sont presque sûrement dans $H_{\mathbb{R}^d}^{k+1}(D)$ et que $\mathbb{P}(\omega)$ -presque sûrement, pour tout $i \in \llbracket 1; d \rrbracket$ et $\alpha \in \mathbb{N}^n$ tel que $|\alpha| \leq k + 1$, $\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X_i(\cdot, \omega) = \partial^\alpha X_i(\cdot, \omega)$ dans $L^2(D, \lambda^n)$. Donc par le corollaire 6.3.1, on dispose de $L \in \mathbb{R}_+^*$ tel que, $\mathbb{P}(\omega)$ -presque sûrement, pour tout $h \in \mathbb{R}_+^*$:

$$\begin{aligned} \|X(\cdot, \omega) - \mathcal{R}_{h,d}(X(\cdot, \omega))\|_{H_{\mathbb{R}^d}^1(D)}^2 &= \|X(\cdot, \omega) - R_{h,d}(X(\cdot, \omega))\|_{H_{\mathbb{R}^d}^1(D)}^2 \\ &\leq L^2 \cdot h^{2k} \cdot \|X(\cdot, \omega)\|_{H_{\mathbb{R}^d}^{k+1}(D)}^2 \end{aligned}$$

De plus $\mathbb{P}(\omega)$ -presque sûrement :

$$\begin{aligned} \|X(\cdot, \omega)\|_{H_{\mathbb{R}^d}^{k+1}(D)}^2 &= \sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \int_D |\partial^\alpha X_i(t, \omega)|^2 dt \\ &= \sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \int_D \left| \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X_i(\cdot, \omega) \right|^2 dt \end{aligned}$$

Donc par passage à l'espérance, pour tout $h \in \mathbb{R}_+^*$:

$$\begin{aligned}
 \mathbb{E}(\|X(\cdot, \omega) - \mathcal{R}_{h,d}(X(\cdot, \omega))\|_{H_{\mathbb{R}^d}^1(D)}^2) &\leq L^2 \cdot h^{2k} \cdot \sum_{i=1}^d \mathbb{E}(\|X(\cdot, \omega)\|_{H_{\mathbb{R}^d}^{k+1}(D)}^2) \\
 &= L^2 \cdot h^{2k} \cdot \sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \mathbb{E}\left(\int_D \left|\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X_i(\cdot, \omega)\right|^2 dt\right) \\
 &= L^2 \cdot h^{2k} \cdot \sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \int_D \mathbb{E}\left(\left|\frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}} X_i(\cdot, \omega)\right|^2\right) dt \\
 &= L^2 \cdot h^{2k} \cdot \sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \int_D D^\alpha C_{i,i}(t, t) dt \\
 &< +\infty
 \end{aligned}$$

Or pour tout $h \in \mathbb{R}_+^*$:

$$\|X - \mathcal{R}_{h,d}X\|^2 \leq \mathbb{E}(\|X(\cdot, \omega) - \mathcal{R}_{h,d}(X(\cdot, \omega))\|_{H_{\mathbb{R}^d}^1(D)}^2)$$

Donc finalement :

$$\|X - \mathcal{R}_{h,d}X\| \leq \left(L \cdot \left(\sum_{i=1}^d \sum_{\substack{\alpha \in \mathbb{N}^n \\ |\alpha| \leq k+1}} \int_D D^\alpha C_{i,i}(t, t) dt \right)^{1/2} \right) \cdot h^k$$

□

Corollaire 6.5.1 (Convergence en loi). Sous les mêmes hypothèses que le théorème 6.5.1 et en supposant de plus que k est un entier non nul alors pour toute fonction $\phi : L_{\mathbb{R}^d}^2(D, \lambda^n) \rightarrow \mathbb{R}$ continue et bornée, on a :

$$\mathbb{E}(\phi(\mathcal{R}_{h,d}X(\cdot, \omega))) \xrightarrow[h>0]{h \rightarrow 0} \mathbb{E}(\phi(X(\cdot, \omega))) \quad (6.11)$$

Démonstration. Soit $(u_n)_{n \in \mathbb{N}}$ une suite de réels strictement positifs tendant vers 0 et $\psi : \mathbb{N} \rightarrow \mathbb{N}$ une extractrice. On a donc que $(u_{\psi(n)})_{n \in \mathbb{N}}$ tend aussi vers 0.

Par le théorème précédent et l'hypothèse que k est non nul, on a que

$$\|X - \mathcal{R}_{h,d}X\| \xrightarrow[h \rightarrow 0]{} 0$$

donc en particulier

$$\mathbb{E}(\|X(\cdot, \omega) - \mathcal{R}_{u_{\psi(n)},d}X(\cdot, \omega)\|_{L_{\mathbb{R}^d}^2(D, \lambda^n)}^2) \xrightarrow[n \rightarrow \infty]{} 0$$

Par conséquent, on dispose de $\theta : \mathbb{N} \rightarrow \mathbb{N}$ une extractrice telle que $\mathbb{P}(\omega)$ -presque sûrement

$$\|X(\cdot, \omega) - \mathcal{R}_{u_{\psi(\theta(n))},d}X(\cdot, \omega)\|_{L_{\mathbb{R}^d}^2(D, \lambda^n)}^2 \xrightarrow[n \rightarrow \infty]{} 0$$

D'où par continuité de ϕ , $\mathbb{P}(\omega)$ -presque sûrement

$$\phi(\mathcal{R}_{u_{\psi(\theta(n))},d}X(.,\omega)) \xrightarrow{n \rightarrow \infty} \phi(X(.,\omega))$$

Comme ϕ est bornée, on peut appliquer le théorème de convergence dominée pour obtenir :

$$\mathbb{E}(\phi(\mathcal{R}_{u_{\psi(\theta(n))},d}X(.,\omega))) \xrightarrow{n \rightarrow \infty} \mathbb{E}(\phi(X(.,\omega)))$$

Finalement on vient de prouver que de toute suite extraite de $(u_n)_{n \in \mathbb{N}}$, $(u_{\phi(n)})_{n \in \mathbb{N}}$, il existe une extractrice θ telle que $\mathbb{E}(\phi(\mathcal{R}_{u_{\psi(\theta(n))},d}X(.,\omega))) \xrightarrow{n \rightarrow \infty} \mathbb{E}(\phi(X(.,\omega)))$.

Par conséquent la suite $(\mathbb{E}(\phi(\mathcal{R}_{u_n,d}X(.,\omega))))_{n \in \mathbb{N}}$ converge et

$$\mathbb{E}(\phi(\mathcal{R}_{u_n,d}X(.,\omega))) \xrightarrow{n \rightarrow \infty} \mathbb{E}(\phi(X(.,\omega)))$$

Comme ce résultat est vrai pour toute suite $(u_n)_{n \in \mathbb{N}}$ de réels strictement positifs tendant vers 0, on a donc prouvé par caractérisation séquentielle le résultat recherché :

$$\mathbb{E}(\phi(\mathcal{R}_{h,d}X(.,\omega))) \xrightarrow[h>0]{h \rightarrow 0} \mathbb{E}(\phi(X(.,\omega)))$$

□

6.6 Erreur et convergence numérique

La méthode de \mathbb{P}_1 -interpolation nécessite la connaissance d'un ouvert $D \subset \mathbb{R}^n$, d'un maillage triangulaire M (notons n_M son nombre de nœuds) discrétisant le domaine \bar{D} et d'un maillage triangulaire d'intérêt M' (notons $n_{M'}$ son nombre de nœuds) dont les simplexes sont inclus dans \bar{D} et d'une fonction de covariance $C : \bar{D} \times \bar{D} \rightarrow \mathbb{R}^d$. Ainsi en utilisant une méthode de simulation en les nœuds de M du champ gaussien X centré et caractérisé par C , on effectue une \mathbb{P}_1 -interpolation à partir d'une réalisation sur M , pour obtenir une réalisation de $\mathcal{R}_{M,d}X$ en les nœuds de M' . Pour les tests, on ne considérera que $n \in \{2, 3\}$ et la méthode de Cholesky sera utilisée pour simuler en les nœuds de M .

6.6.1 Dimension $n=2$

On choisit $\bar{D} = [-1, 1]^2$ et pour $m \in \mathbb{N}^*$, $m > 1$, le maillage M sera caractérisé par des nœuds du type

$$\left(\left(-1 + \frac{2i}{m-1}, -1 + \frac{2j}{m-1} \right) \right)_{(i,j) \in \llbracket 0; m-1 \rrbracket^2}$$

et leurs éléments triangulaires sont décrits de la même façon qu'à la sous-section 2.3.2 du chapitre 2. Si on note h_M le maximum des diamètres des

éléments triangulaires qui composent le maillage M , $h_M = \sqrt{2} \cdot \frac{2}{m-1}$.

On fera varier le nombre de nœuds $n_M = m^2$ entre 10 et 10000 et on ne considérera qu'une seule fonction de covariance C :

$$C(x, y) = \exp\left(-\left(\frac{y_1 - x_1}{b_1}\right)^2 - \left(\frac{y_2 - x_2}{b_2}\right)^2\right)$$

pour $(x, y) = ((x_1, x_2), (y_1, y_2)) \in (\bar{D})^2$.

Le maillage M' est défini en deux temps. D'abord on discrétise le domaine \bar{D} par un maillage BE dont on note n_{BE} le nombre de nœuds. Pour $m_{BE} \in \mathbb{N}^*$, $m_{BE} > 1$, le maillage BE sera caractérisé par des nœuds du type

$$\left(\left(-1 + \frac{2i}{m_{BE} - 1}, -1 + \frac{2j}{m_{BE} - 1}\right)\right)_{(i,j) \in \llbracket 0; m_{BE}-1 \rrbracket^2}$$

et leurs éléments triangulaires sont décrits de la même façon qu'à la sous-section 2.3.2 du chapitre 2. Le nombre de nœuds $n_{BE} = m_{BE}^2$ est fixé à 784. On pose ensuite $f : (x, y) \rightarrow 1 - \sqrt{x^2 + y^2}$ et $v = 1 - \frac{1}{\sqrt{2}}$. On obtient alors le maillage d'intérêt M' en parcourant les éléments triangulaires et en raisonnant de la façon suivante : si tous les sommets s de l'élément vérifient $f(s) > v$ alors on enlève cet élément du maillage, sinon si un sommet s (mais pas tous) vérifie $f(s) > v$ alors on modifie la position du sommet s de façon à ce qu'il vérifie $f(s) = v$. Le maillage M' ainsi obtenu est un maillage du disque de centre 0 et de rayon $v = \frac{1}{\sqrt{2}}$.

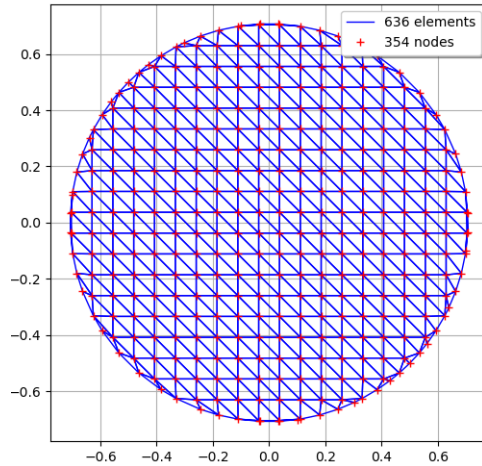


FIGURE 6.1 – Maillage d'intérêt M' en dimension 2

6.6.2 Dimension $n=3$

On choisit $\bar{D} = [-1, 1]^3$ et pour $m \in \mathbb{N}^*$, $m > 1$, le maillage M sera caractérisé par des nœuds du type

$$\left(\left(-1 + \frac{2i}{m-1}, -1 + \frac{2j}{m-1}, -1 + \frac{2k}{m-1} \right) \right)_{(i,j,k) \in \llbracket 0; m-1 \rrbracket^3}$$

et leurs éléments tétraédriques sont décrits de la même façon qu'à la sous-section 2.3.3 du chapitre 2. Si on note h_M le maximum des diamètres des éléments triangulaires qui composent le maillage M , $h_M = \sqrt{3} \cdot \frac{2}{m-1}$.

Le nombre de nœuds $n_M = m^3$ variera entre 10 et 10000 et on ne considérera qu'une seule fonction de covariance C :

$$C(x, y) = \exp(-\|y - x\|_2^2) \text{ pour } (x, y) \in (\bar{D})^2$$

Le maillage M' est défini en deux temps. D'abord on discrétise le domaine \bar{D} par un maillage BE dont on note n_{BE} le nombre de nœuds. Pour $m_{BE} \in \mathbb{N}^*$, $m_{BE} > 1$, le maillage BE sera caractérisé par des nœuds du type

$$\left(\left(-1 + \frac{2i}{m_{BE}-1}, -1 + \frac{2j}{m_{BE}-1}, -1 + \frac{2k}{m_{BE}-1} \right) \right)_{(i,j,k) \in \llbracket 0; m_{BE}-1 \rrbracket^3}$$

et leurs éléments tétraédriques sont décrits de la même façon qu'à la sous-section 2.3.3 du chapitre 2.

Le nombre de nœuds $n_{BE} = m_{BE}^3$ est fixé à 1000. On pose ensuite la fonction $f : (x, y, z) \rightarrow 1 - \sqrt{x^2 + y^2 + z^2}$ et $v = 1 - \frac{1}{\sqrt[3]{2}}$. On obtient alors le maillage d'intérêt M' en parcourant les éléments tétraédriques et en raisonnant de la façon suivante : si tous les sommets s de l'élément vérifient $f(s) > v$ alors on enlève cet élément du maillage, sinon si un sommet s (mais pas tous) vérifie $f(s) > v$ alors on modifie la position du sommet s de façon à ce qu'il vérifie $f(s) = v$. Le maillage M' ainsi obtenu est un maillage de la boule de centre 0 et de rayon $v = \frac{1}{\sqrt[3]{2}}$.

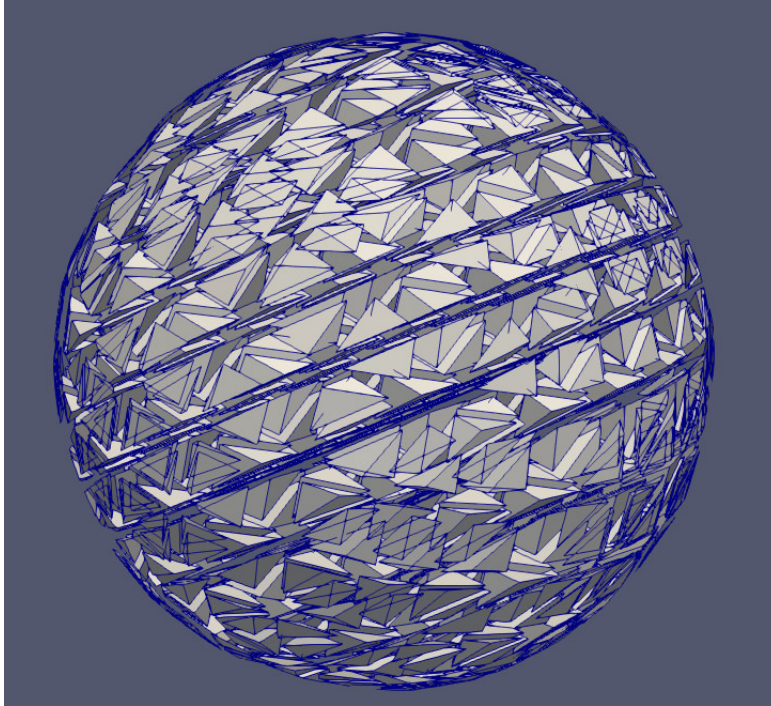


FIGURE 6.2 – Maillage d'intérêt M' en dimension 3

6.6.3 Benchmarks

Dimension	Nb de nœuds de M'	Nb de nœuds de M n_M	Temps moyen d'une réalisa- tion (en seconde)
2	354	9	0.00457s
2	354	100	0.00558s
2	354	961	0.11667s
2	354	10000	146.70s
3	474	8	0.01782s
3	474	64	0.02263s
3	474	729	0.09118s
3	474	9261	117.63s

Dimension	Nb de nœuds de M'	Nb de nœuds de M n_M	h_M	Nb de réalisa- tions	erreur L^2
2	354	16	0.94280	250	0.09704
2	354	16	0.94280	1000	0.13483
2	354	16	0.94280	4000	0.11302
2	354	16	0.94280	10000	0.11756
2	354	121	0.28284	250	0.08694
2	354	121	0.28284	1000	0.02646
2	354	121	0.28284	4000	0.01703
2	354	121	0.28284	10000	0.01040
2	354	529	0.12856	250	0.16377
2	354	529	0.12856	1000	0.01925
2	354	529	0.12856	4000	0.01875
2	354	529	0.12856	10000	0.00689
2	354	1024	0.09123	250	0.04241
2	354	1024	0.09123	1000	0.01526
2	354	1024	0.09123	4000	0.04384
2	354	1024	0.09123	10000	0.01988
3	474	27	1.7320	250	0.43154
3	474	27	1.7320	1000	0.43918
3	474	27	1.7320	4000	0.45128
3	474	27	1.7320	10000	0.45764
3	474	125	0.86602	250	0.17489
3	474	125	0.86602	1000	0.18455
3	474	125	0.86602	4000	0.17802
3	474	125	0.86602	10000	0.15349
3	474	512	0.49487	250	0.14222
3	474	512	0.49487	1000	0.07678
3	474	512	0.49487	4000	0.07379
3	474	512	0.49487	10000	0.06428
3	474	1000	0.38490	250	0.11127
3	474	1000	0.38490	1000	0.05500
3	474	1000	0.38490	4000	0.03705
3	474	1000	0.38490	10000	0.02903

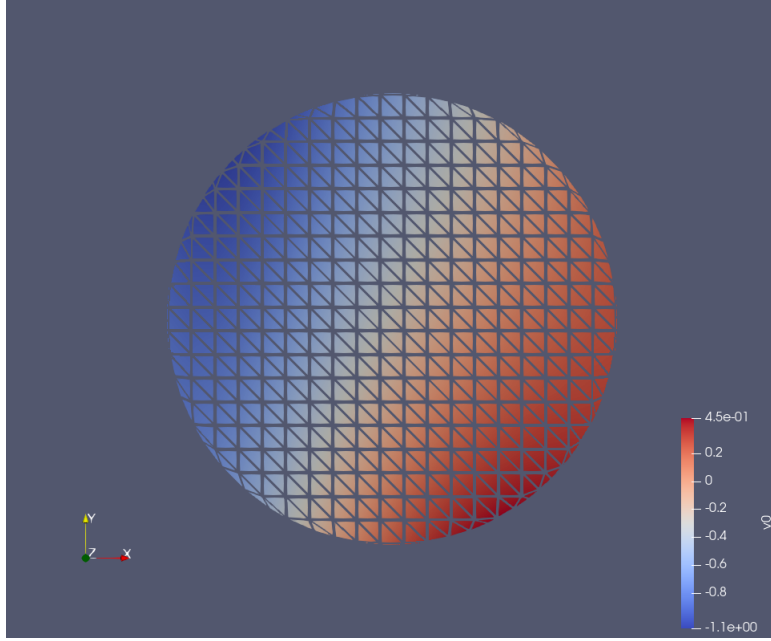


FIGURE 6.3 – Réalisation sur M' en dimension 2 où $n_M = 961$

À des fins de comparaison, on rajoute ci-dessous un benchmark où on applique directement la méthode de Cholesky sur le maillage d'intérêt M' . On ne considère donc plus le maillage M .

Dimension	Nb de nœuds de M'	Nb de réalisations	erreur L^2
2	354	250	0.04956
2	354	1000	0.01539
2	354	4000	0.01319
2	354	10000	0.01148
3	474	250	0.17656
3	474	1000	0.08154
3	474	4000	0.03933
3	474	10000	0.01620

Bibliographie

- [1] Grégoire ALLAIRE : *Analyse numérique et optimisation : une introduction à la modélisation mathématique et à la simulation numérique*. Mathématiques appliquées. Éditions de l'École Polytechnique, Palaiseau, 2e édition édition, 2012.
- [2] BERNARD, FOGLI et POIRION : Simulation des processus stochastiques. Cours du Collège de l'X, 2012.
- [3] Carlo GAETAN et Xavier GUYON : *Spatial Statistics and Modeling*. Springer Series in Statistics. Springer Nature, New York, NY, 2009.
- [4] Alain GALLI et Haiyu GAO : Rate of convergence of the gibbs sampler in the gaussian case. *Mathematical geology*, 33(6):653–677, 2001.
- [5] GIKHMAN et SKOROKHOD : *The Theory of Stochastic Processes I*. Classics in Mathematics. Springer Berlin, Heidelberg, 2004.
- [6] Dionissios T. HRISTOPULOS : *Random Fields for Spatial Data Modeling : A Primer for Scientists and Engineers*. Advances in Geographic Information Science. Springer Netherlands, Dordrecht, 1st ed. 2020. édition, 2020.
- [7] Frédéric KLOPP : Analyse réelle et analyse harmonique. <https://webusers.imj-prg.fr/~frederic.klopp/cours/m1-4m030-21-22/AnalyseReelleAnalyseHarmoniqueDistributions.pdf>.
- [8] Christian LANTUÉJOUL et Nicolas DESASSIS : Simulation of a gaussian random vector : A propagative version of the gibbs sampler. 2012.
- [9] Benoît LIZÉ : Résolution directe rapide pour les éléments finis de frontière en électromagnétisme et acoustique : H-matrices. Parallélisme et applications industrielles, 2014.
- [10] Sergei PRIGARIN : *Spectral Models of Random Fields in Monte Carlo Methods*. 2001.
- [11] Michael SCHEUERER : Regularity of the sample paths of a general second order random field. *Stochastic processes and their applications*, 120(10):1879–1897, 2010.
- [12] Christian SOIZE : *Méthodes mathématiques en analyse du signal*. Enseignement de la physique. Masson, Paris, 1993.

Annexe A

Base théorique pour la méthode spectrale

On introduit ici les outils théoriques permettant de comprendre la méthode spectrale (voir chapitre 5). Pour la section A.1, les terminologies de mesure complexe et de variation totale d'une mesure complexe ont été puisées dans [7].

A.1 Mesure matricielle

Soit (Λ, ξ) un espace mesurable, $d \in \mathbb{N}^*$, $\mu = (\mu_{i,j})_{(i,j) \in \llbracket 1; d \rrbracket^2}$ une mesure vectorielle sur (Λ, ξ) à valeurs dans $M_d(\mathbb{C})$ où les $\mu_{i,j}$ sont des mesures complexes dont on note leur variation totale $|\mu_{i,j}|$.

Pour $p \in [1, \infty[$, on notera $L^p(\Lambda, \mu)$ l'ensemble des fonctions mesurables $f : \Lambda \rightarrow \mathbb{C}$ telles que $\forall (i, j) \in \llbracket 1; d \rrbracket^2, f \in L^p(\Lambda, |\mu_{i,j}|)$. L'une des conséquences immédiates est que pour $f \in L^1(\Lambda, \mu)$:

$$\int_{\Lambda} f d\mu = \begin{pmatrix} \int_{\Lambda} f d\mu_{1,1} & \cdots & \int_{\Lambda} f d\mu_{1,d} \\ \vdots & \ddots & \vdots \\ \int_{\Lambda} f d\mu_{d,1} & \cdots & \int_{\Lambda} f d\mu_{d,d} \end{pmatrix} \in M_d(\mathbb{C}) \text{ est bien définie}$$

De plus on remarquera que comme la variation totale d'une mesure complexe est une mesure positive finie, pour $(p, q) \in [1, \infty]^2$ tel que $p \leq q$, $L^q(\Lambda, \mu) \subset L^p(\Lambda, \mu)$. D'où en particulier $L^2(\Lambda, \mu) \subset L^1(\Lambda, \mu)$.

Pour $p \in [1, \infty[$, on peut noter aussi que $L^p(\Lambda, \mu) = L^p(\Lambda, \nu)$ où ν est une mesure positive sur (Λ, ξ) telle que $\forall A \in \xi, \nu(A) = \sum_{(i,j) \in \llbracket 1; d \rrbracket^2} |\mu_{i,j}|(A)$.

D'où une structure d'espace de Banach de style L^p telle que l'on connaît bien. Pour la suite, ν sera notée $\|\mu\|$.

A.2 Mesure stochastique orthogonale centrée et isométrie

Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé, $\mu = (\mu_{i,j})_{(i,j) \in \llbracket 1;d \rrbracket^2}$ une mesure vectorielle sur (Λ, ξ) à valeurs dans $M_d(\mathbb{C})$ où les $\mu_{i,j}$ sont des mesures complexes.

On note :

- $\langle \cdot, \cdot \rangle$ l'application telle que $\forall (f, g) \in L^2(\Lambda, \mu)^2$, $\langle f, g \rangle = \int_{\Lambda} f \bar{g} d\mu$
- $L_0^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$, les vecteurs aléatoires centrés de $L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$
- $\|\cdot\|_{L^2(\mathbb{P}; \mathbb{C}^d)}$ la norme hilbertienne de $L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$ telle que $\|X\|_{L^2(\mathbb{P}; \mathbb{C}^d)}^2$ vaut $\mathbb{E}(\|X\|_2^2)$, où $\|\cdot\|_2$ est la norme hermitienne naturelle dans \mathbb{C}^d

Remarque A.2.1. $L_0^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$ est un sous-espace vectoriel fermé de $L^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$.

Définition A.2.1. Soit $I : L^2(\Lambda, \mu) \rightarrow L_0^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$ une application. On dit que I est une mesure stochastique orthogonale centrée (m.s.o.c) de mesure structurelle (ou de base) μ si $\forall f \in L^2(\Lambda, \mu)$:

$$\langle f, f \rangle = \mathbb{E}(I(f)I(f)^*) \quad (\text{A.1})$$

Et si tel est le cas, on définit la mesure stochastique associée à I comme l'application $\phi : \xi \rightarrow L_0^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$ telle que $\forall B \in \xi$, $\phi(B) = I(\mathbb{1}_B)$.

Remarque A.2.2.

$$\begin{aligned} \forall f \in L^2(\Lambda, \mu), \langle f, f \rangle &= \mathbb{E}(I(f)I(f)^*) \\ &\iff \\ \forall (f, g) \in L^2(\Lambda, \mu)^2, \langle f, g \rangle &= \mathbb{E}(I(f)I(g)^*) \end{aligned} \quad (\text{A.2})$$

Supposons maintenant que I est une m.s.o.c. D'où les propriétés suivantes :

- $\text{tr}(\mu) = \sum_{i=1}^d \mu_{i,i}$ est une mesure positive sur (Λ, ξ) (mesure-trace de μ)
- $\forall (i, j) \in \llbracket 1; d \rrbracket^2$, $\forall (f, g) \in L^2(\Lambda, \mu)^2$:
 $|\int_{\Lambda} f \bar{g} d\mu_{i,j}| \leq (\int_{\Lambda} |f|^2 d\text{tr}(\mu))^{\frac{1}{2}} (\int_{\Lambda} |g|^2 d\text{tr}(\mu))^{\frac{1}{2}}$

- $\forall A \in \xi, |\mu_{i,j}(A)| \leq |\mu_{i,j}|(A) \leq \text{tr}(\mu)(A)$ (la mesure-trace contrôle tous les coefficients de la mesure vectorielle μ)
- $L^2(\Lambda, \mu) = L^2(\Lambda, \text{tr}(\mu))$ et I est une isométrie de $L^2(\Lambda, \text{tr}(\mu))$ dans $L_0^2(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{C}^d)$
- ϕ la mesure stochastique associée à I est bien une mesure stochastique sur (Λ, ξ) à valeurs dans $L_0^2(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{C}^d)$ i.e que pour toute famille dénombrable de parties mesurables 2 à 2 disjointes $(A_j)_{j \in \mathbb{N}}$,
 $\phi(\cup_{j \in \mathbb{N}} A_j) = \sum_{j \in \mathbb{N}} \phi(A_j)$ (somme au sens de la norme $\|\cdot\|_{L^2(\mathbb{P}, \mathbb{C}^d)}$)

Définition A.2.2. Soit $I : L^2(\Lambda, \mu) \rightarrow L_0^2(\Omega, \mathcal{A}, \mathbb{P}; \mathbb{C}^d)$ une m.s.o.c de mesure de base μ dont on note ϕ la mesure stochastique associée. On définit pour $f \in L^2(\Lambda, \mu)$, l'intégrale de Wiener de f associée à la mesure stochastique ϕ ainsi :

$$\int_{\Lambda} f(\lambda) \phi(d\lambda) = I(f) \quad (\text{A.3})$$

Et pour toute partie mesurable $\Gamma \subset \Lambda$, pour tout $f \in L^2(\Lambda, \mu)$, on définit :
 $\int_{\Gamma} f(\lambda) \phi(d\lambda) = \int_{\Lambda} \mathbf{1}_{\Gamma}(\lambda) f(\lambda) \phi(d\lambda) = I(\mathbf{1}_{\Gamma} f)$

Remarque A.2.3. Si on a une mesure stochastique ϕ_2 issue d'une autre m.s.o.c I_2 de mesure de base aussi μ , alors pour démontrer que $\phi_2 = \phi$, il suffit juste de démontrer que l'on dispose de A une partie dense de $L^2(\Lambda, \text{tr}(\mu))$ telle que l'intégrale de Wiener associée à ϕ et celle associée à ϕ_2 sont égales pour tout élément de A (ce qui revient à dire $I|_A = (I_2)|_A$)

A.3 Sur l'existence d'une m.s.o.c associée à un champ aléatoire

Pour plus de détails sur les résultats généraux de cette section, voir [12].

Considérons (Ω, F, \mathbb{P}) un espace probabilisé, n et d deux entiers naturels non nuls et $X : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^d$ un champ aléatoire. On émet l'hypothèse suivante :

Hypothèse A.3.1. X est d'ordre 2, centré, continue en moyenne quadratique, faiblement stationnaire d'ordre 2.

On note $R : \mathbb{R}^n \rightarrow M_d(\mathbb{C})$ la fonction d'autocovariance de X .

A.3.1 Mesure spectrale

Théorème A.3.1. Sous l'hypothèse A.3.1, on dispose d'une unique mesure vectorielle $\mu = (\mu_{i,j})_{(i,j) \in \llbracket 1;d \rrbracket^2}$ sur $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$ à valeurs dans l'ensemble des matrices hermitiennes semi-définies positives de $M_d(\mathbb{C})$, dont les $\mu_{i,j}$ sont des mesures complexes et telle que :

$$\forall t \in \mathbb{R}^n, R(t) = \int_{\mathbb{R}^n} \exp(i \langle t, \omega \rangle_n) d\mu(\omega) \quad (\text{A.4})$$

($\langle \cdot, \cdot \rangle_n$ est le produit scalaire usuel dans \mathbb{R}^n)

Définition A.3.1. Avec les notations du précédent théorème, on appelle μ la mesure spectrale matricielle ou juste mesure spectrale de X et on la note M_X .

Propriété A.3.1. $tr(M_X)$ est une mesure positive sur $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n))$.

Propriété A.3.2. $\forall A \in \mathcal{B}(\mathbb{R}^n), M_X(A) = \overline{M_X(-A)}$ donc en particulier $tr(M_X)(A) = tr(M_X)(-A)$.

A.3.2 Densité spectrale

Définition A.3.2. Si les mesures $(M_X)_{i,j}$ sont à densité par rapport à la mesure de Lebesgue $d\omega$ de \mathbb{R}^n alors on dispose de $S = (S_{i,j})_{(i,j) \in \llbracket 1;d \rrbracket^2}$ une application mesurable allant de \mathbb{R}^n dans $M_d(\mathbb{C})$, $d\omega$ -intégrable et unique à un ensemble de mesure nulle près par rapport à la mesure $d\omega$, telle que :

$$\forall A \in \mathcal{B}(\mathbb{R}^n), M_X(A) = \int_A S(\omega) d\omega \quad (\text{A.5})$$

$$\text{Donc } \forall (i, j) \in \llbracket 1;d \rrbracket^2, \forall A \in \mathcal{B}(\mathbb{R}^n), (M_X)_{i,j}(A) = \int_A S_{i,j}(\omega) d\omega$$

On appelle S la densité spectrale de X et on la note S_X .

Supposons jusqu'à la fin de cette sous-section que le champ X admet une densité spectrale S_X . On observe alors ces propriétés :

- $\forall t \in \mathbb{R}^n, R(t) = \int_{\mathbb{R}^n} \exp(i \langle t, \omega \rangle_n) S_X(\omega) d\omega$
 - S_X est $d\omega$ -presque partout hermitienne semi-définie positive
 - Pour presque tout $\omega \in \mathbb{R}^n$, $S_X(-\omega) = \overline{S_X(\omega)}$
 - Si $\|R\|_F$ est dans $L^1(\mathbb{R}^n, dt)$ (où $dt = d\omega$) alors S_X peut être décrite explicitement via une transformée de Fourier :
- $$\forall \omega \in \mathbb{R}^n, S_X(\omega) = (2\pi)^{-n} \int_{\mathbb{R}^n} \exp(-i \langle \omega, t \rangle_n) R(t) dt.$$

Remarque A.3.1. Tous les coefficients de R sont dans $L^1(\mathbb{R}^n, dt)$ si et seulement si $\|R\|_F$ est dans $L^1(\mathbb{R}^n, dt)$ ($\|\cdot\|_F$ désigne la norme de Frobenius).

A.3.3 Densité spectrale fréquentielle

Mettons en avant maintenant le fait que la notation ω souligne la notion de pulsation rencontrée en physique mais il est possible que l'on souhaite parler non pas de pulsation mais de fréquence $f = \frac{\omega}{2\pi}$. C'est pourquoi l'on définit une autre densité spectrale que l'on appelle densité spectrale fréquentielle. On suppose de nouveau que le champ aléatoire X admet une densité spectrale S_X .

Définition A.3.3. On définit alors la densité spectrale fréquentielle de X comme l'application $S_X^{fr} : \mathbb{R}^n \rightarrow M_d(\mathbb{R})$ telle que :

$$\forall f \in \mathbb{R}^n, S_X^{fr}(f) = (2\pi)^n S_X(2\pi f)$$

Propriété A.3.3. Si $\|R\|_F$ est dans $L^1(\mathbb{R}^n, dt)$ alors S_X^{fr} peut être totalement décrite par une transformée de Fourier de la fonction d'autocovariance R :

$$\forall f \in \mathbb{R}^n, S_X^{fr}(f) = \int_{\mathbb{R}^n} \exp(-2\pi i \langle f, t \rangle_n) R(t) dt$$

A.4 Représentation spectrale de X

Sous l'hypothèse A.3.1, on a pu exhiber dans la section précédente la notion de mesure spectrale matricielle mais on peut aller plus loin et démontrer l'existence d'une unique mesure stochastique associée à X , ϕ_X , issue d'une unique m.s.o.c. I_X décrivant totalement X . Précisément sous l'hypothèse A.3.1 :

Théorème A.4.1. Il existe une unique mesure stochastique $\phi_X : \mathcal{B}(\mathbb{R}^n) \rightarrow L_0^2(\Omega, \mathcal{A}, \mathbb{P}, \mathbb{C}^d)$ associée à une unique m.s.o.c. I_X de mesure de base la mesure spectrale M_X , telle que :

$$\forall t \in \mathbb{R}^n, X(t, \cdot) = \int_{\mathbb{R}^n} \exp(i \langle t, \omega \rangle_n) \phi_X(d\omega)$$

Si de plus X est un champ gaussien, ϕ_X est une mesure stochastique gaussienne centrée i.e $\forall A \in \mathcal{B}(\mathbb{R}^n)$, $\phi_X(A)$ est un vecteur gaussien à valeurs dans \mathbb{C}^d , centré, dont la matrice de covariance complexe vaut $\mathbb{E}(\phi_X(A)\phi_X(A)^*) (= M_X(A))$

Propriété A.4.1. $\forall f \in L^2(\mathbb{R}^n, M_X)$, $\overline{I_X(f)} = I_X(\bar{f}(-\cdot))$

Remarque A.4.1. Cette première propriété est inspirée de la page 52 du livre de Prigarin [10].

Propriété A.4.2. $\forall A \in \mathcal{B}(\mathbb{R}^n)$, $\phi_X(A) = \overline{\phi_X(-A)}$

Propriété A.4.3. $\forall (A, B) \in \mathcal{B}(\mathbb{R}^n)^2$, $\mathbb{E}(\phi_X(A)\phi_X(B)^T) = M_X(A \cap (-B))$

Propriété A.4.4. Si X est un champ gaussien alors toute famille d'éléments de $I_X(L^2(\mathbb{R}^n, M_X))$ indexée par un ensemble D quelconque est un processus gaussien centré.

Annexe B

Code pour les tests numériques

Du code utilisant la librairie OpenTURNS a été créé pour les tests numériques. Ce code est disponible sous demande sur un dépôt git (Github). On se propose donc ici de décrire la structure des dossiers/fichiers de ce dépôt. Tout est rassemblé dans un même dossier de nom Benchmarks.

B.1 Dossier CholeskyMethod

Le dossier est composé de deux fichiers. Le fichier testTool.py contient une unique fonction du nom checkCovariance() qui consiste à partir un échantillon de réalisations de déterminer l'erreur L^2 (voir la sous-section 2.2.1 du chapitre 2) par rapport à une matrice Σ construite de la même façon qu'au chapitre 1 à l'aide du maillage où on effectue les réalisations et d'une fonction de covariance C . Ce fichier existe dans tous les dossiers qui seront décrits dans les sections suivantes mais pour le dossier SpectralMethod, le fichier testTool.py sera agrémenté d'autres fonctions. Le second fichier se nomme cholesky.py et il contient des tests numériques qui sont détaillés au chapitre 2 à la section 2.3.

B.2 Dossier HmatrixMethod

Le dossier est composé de deux fichiers. Le fichier testTool.py et le fichier Hmatrix.py qui contient des tests numériques qui sont détaillés au chapitre 3 à la section 3.4.

B.3 Dossier GalliGaoGibbsMethod

Le dossier est composé de deux fichiers. Le fichier testTool.py et le fichier GGG.py qui contient des tests numériques qui sont détaillés au chapitre 4 à la section 4.4.

B.4 Dossier SpectralMethod

Le dossier est composé de 11 fichiers et par la suite, quand on parlera de densité spectrale, on sous-entendra densité spectrale fréquentielle (voir la sous-section A.3.3 de l'annexe A). Les fichiers `mySpectralGaussianProcess1D.py`, `mySpectralGaussianProcess2D.py` et `mySpectralGaussianProcess3D.py` contiennent l'implémentation de la méthode spectrale en fonction de la dimension d'entrée du processus $X : \mathbb{R}^n \times \Omega \rightarrow \mathbb{R}^d$ ($n=1, 2$ ou 3). Ces fichiers ne contiennent qu'une classe, celle qui implémente la méthode spectrale et qui porte le même nom que le fichier sauf que ça commence par une majuscule. La classe, en plus du constructeur, contient systématiquement les méthodes utilisables suivantes : `getRealization()`, `getSample()`, `getOutputDimension()` et `getMesh()`.

Les fichiers `estimationSpectraleDim1.py`, `estimationSpectraleDim2.py`, `estimationSpectraleDim3.py` contiennent l'implémentation de l'estimateur de la densité spectrale évoquée à la section 5.9 du chapitre 5. Ces fichiers ne contiennent qu'une classe, celle qui implémente l'estimateur de la densité spectrale. La classe, en plus du constructeur, contient systématiquement la méthode utilisable `buildFromSample()` qui évalue l'estimateur en fonction des réalisations sur le maillage formé par les points de simulation et (en utilisant les notations des sections 5.3 et 5.4 du chapitre 5) en fonction de N_1, \dots, N_n et de $T_1 - \Delta t_1, \dots, T_n - \Delta t_n = (N_n - 1)\Delta t_n$. Plus explicitement, la syntaxe se présente ainsi :

`buildFromSample(sample, [N1, ..., Nn], [T1 - Δt1, ..., Tn - Δtn])`

Le fichier `spectralModels.py` contient l'implémentation de plusieurs densités spectrales :

$$S(f) = \frac{1}{2A} \mathbb{1}_{[-A, A]}(f) \quad (S1)$$

$$S(f) = S(f_1, \dots, f_n) = \sigma^2 (\pi\theta)^{\frac{n}{2}} \exp(-\pi^2 \theta \|f\|_2^2) \cdot I_d \quad (S2)$$

$$S(f) = S(f_1, f_2, f_3) = \frac{8\pi a}{(a^2 + (2\pi)^2 \|f\|_2^2)^2} \quad (S3)$$

$$S(f) = S(f_1, f_2) = \sigma^2 b_1 \cdot b_2 \cdot \pi \cdot \exp(-\pi^2 (b_1^2 f_1^2 + b_2^2 f_2^2)) \quad (S4)$$

Les fonctions d'autocovariance associées à ces densités sont respectivement :

$$R(t) = \frac{\sin(2\pi A t)}{2\pi A t} \quad (R1)$$

$$R(t) = R(t_1, \dots, t_n) = \sigma^2 \exp(-\frac{\|t\|_2^2}{\theta}) \cdot I_d \quad (R2)$$

$$R(t) = R(t_1, t_2, t_3) = \exp(-a\|t\|_2) \quad (R3)$$

$$R(t) = R(t_1, t_2) = \sigma^2 \exp(-(t_1/b_1)^2 - (t_2/b_2)^2) \quad (R4)$$

Les densités (S1), (S3), et (S4) sont implémentées par les classes respectives `UniformSpectralModel`, `SpectralModelofExponential3D1D` et `SpectralModelofAnExponential2D1D`. La densité (S2) est implémentée de façon générale par la classe `GaussianSpectralModel` mais dans le cas où $n = 1$, on utilise la classe `GaussianSpectralModelDim1` afin d'éviter des problèmes de compatibilité avec la librairie `OpenTURNS`.

Le fichier `tool.py` contient deux méthodes. Toujours en utilisant les notations des sections 5.3 et 5.4 du chapitre 5, la méthode `createSpatialSteps()` permet de déduire la liste $[\Delta t_1, \dots, \Delta t_n]$ à partir de la liste $[N_1 - 1, \dots, N_n - 1]$ et la liste $[T_1 - \Delta t_1, \dots, T_n - \Delta t_n]$. En effet pour $i \in \llbracket 1; n \rrbracket$, $\Delta t_i = \frac{T_i - \Delta t_i}{N_i - 1}$. Plus explicitement, la syntaxe se présente ainsi :

$$[\Delta t_1, \dots, \Delta t_n] = \text{createSpatialSteps}([N_1 - 1, \dots, N_n - 1], [T_1 - \Delta t_1, \dots, T_n - \Delta t_n])$$

La méthode `createSpectralSteps()` permet de déduire la liste $[\Delta f_1, \dots, \Delta f_n]$ à partir de la liste $[T_1, \dots, T_n]$. En effet pour $i \in \llbracket 1; n \rrbracket$, $\Delta f_i = \frac{1}{T_i}$. La syntaxe se présente ainsi :

$$[\Delta f_1, \dots, \Delta f_n] = \text{createSpectralSteps}([T_1, \dots, T_n])$$

Le fichier `userDefinedSpectralModel.py` contient une classe de même nom (à une majuscule près) ainsi qu'une fonction utile à la classe. Cette classe permet de définir une densité spectrale S sur un pavé $D_f = [-f_{\max,1}, f_{\max,1}] \times \dots \times [-f_{\max,n}, f_{\max,n}]$. Chaque arête $[-f_{\max,j}, f_{\max,j}]$ a été subdivisée en N_j intervalles de longueur Δf_j de centres respectifs $f_{j,1}, \dots, f_{j,N_j}$. La densité spectrale S est définie sur D_f par morceaux et est constante sur les pavés

$$\tilde{M}_{(k_1, \dots, k_n)} = [f_{1,k_1} - \frac{\Delta f_1}{2}, f_{1,k_1} + \frac{\Delta f_1}{2}] \times \dots \times [f_{n,k_n} - \frac{\Delta f_n}{2}, f_{n,k_n} + \frac{\Delta f_n}{2}]$$

où $(k_1, \dots, k_n) \in \llbracket 1; N_1 \rrbracket \times \dots \times \llbracket 1; N_n \rrbracket$.

Le fichier `testTool.py` contient de multiples fonctions. On a la fonction `checkCovariance()` évoquée à la section B.1. On rajoute ensuite la fonction `createStationaryCovarianceModel()` qui à partir d'un maillage M donné et d'une fonction d'autocovariance R allant de \mathbb{R}^n dans \mathbb{R} (d , la dimension de sortie vaut ici 1) crée un modèle gaussien de covariance sur le maillage M , c'est-à-dire en utilisant les notations du chapitre 1 (où m désigne le nombre de nœuds du maillage) que l'on simule le vecteur gaussien $\mathcal{N}(0_{\mathbb{R}^{dm}}, \Sigma)$ où pour n_i et n_j deux nœuds du maillage, on a $\Sigma_{i,j} = C(n_i, n_j) = R(n_j - n_i)$. Ensuite pour $K \in \llbracket 1; 3 \rrbracket$, on dispose des fonctions suivantes : `getNormInfDimK()`

et `compareSpectralModelsNormInfDimK()`. Les deux fonctions nécessitent une famille F de points de l'espace descriptibles de la même façon que les points

$$((f_{1,k_1}, \dots, f_{n,k_n}))_{k \in \llbracket 1; N_1 \rrbracket \times \dots \times \llbracket 1; N_n \rrbracket}$$

évoqués plus tôt pour le fichier `userDefinedSpectralModel.py`. Pour `getNormInfDimK()`, avec l'information d'une densité spectrale S , la fonction va chercher le point dont la norme de Frobenius de son image par S est la plus élevée et retournera cette valeur. Pour la fonction `compareSpectralModelsNormInfDimK()` avec l'information de deux densités spectrales S_1 et S_2 , la fonction va chercher le point dont la norme de Frobenius de son image par $S_2 - S_1$ est la plus élevée et retournera cette valeur. C'est donc cette fonction que l'on utilisera pour évaluer l'erreur entre deux densités spectrales en les points de la famille F . La dernière fonction s'y trouvant est la fonction `graphOfTwoModelsInputDim1OutputDim1()` qui va dessiner le graphe de deux densités spectrales allant de \mathbb{R} dans \mathbb{R} dont on considère les valeurs pour les points de F .

Le dernier fichier à évoquer est le fichier qui décrit des tests numériques : `spectral.py`. Le détail des tests effectués se trouvent dans la section 5.9 du chapitre 5.

B.5 Dossier P1InterpolationMethod

Le dossier est composé de trois fichiers. Le fichier `testTool.py`, le fichier `P1Interpolation.py` qui contient des tests numériques qui sont détaillés au chapitre 6 à la section 6.6 et le fichier `P1InterpolationGaussianProcess.py`. Ce dernier fichier contient la classe qui implémente la méthode par \mathbb{P}_1 -interpolation. Cette classe de nom `P1InterpolationGaussianProcess` contient, en plus du constructeur, la méthode `getRealization()`, `getSample()`, `getMesh()`, `getOutputDimension()`, `changeEnvelopingProcess()` et `getEnvelopingMesh()`. Ici ce que l'on désigne comme `EnvelopingProcess` désigne le champ gaussien simulé au niveau des nœuds du maillage `EnvelopingMesh` qui n'est rien d'autre que le maillage M évoqué au début de la section 6.6 du chapitre 6.