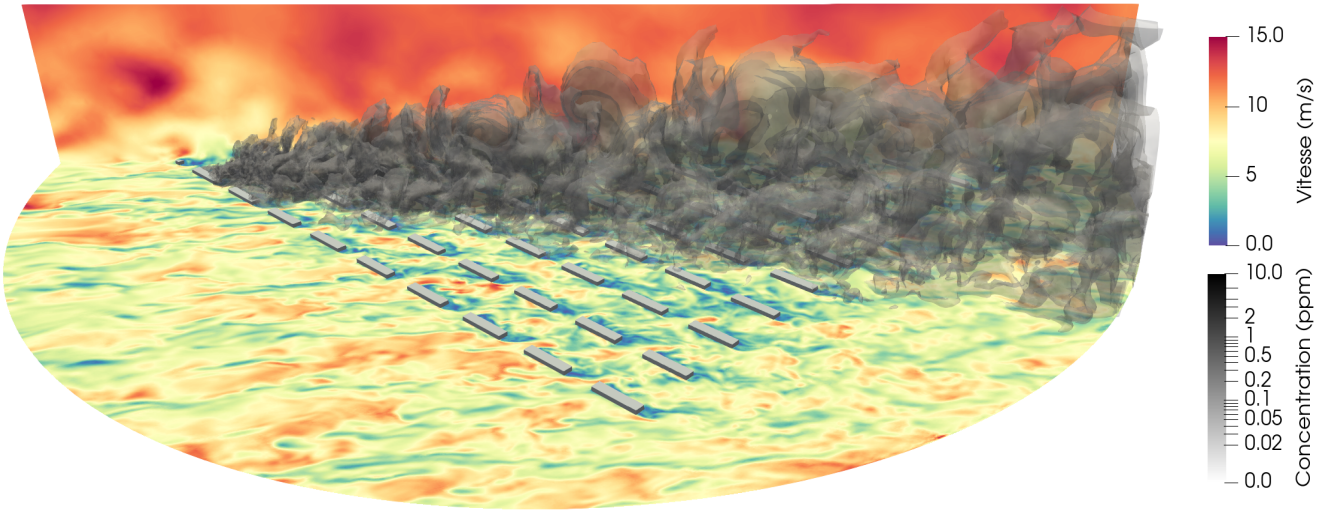
RAPPORT DE STAGE

Méthodes de réduction de dimension par apprentissage pour l'émulation statistique d'un modèle de dispersion atmosphérique



Simulation aux grandes échelles de tourbillons de l'écoulement atmosphérique et de la dispersion d'un polluant dans une canopée urbaine simplifié par [Lumet, 2024]

DELAR Emmarius

Université de Rennes
Faculté des Sciences Économiques
Master 1 Mathématiques Appliquées, Statistique
Promotion 23/24
Du 2 Mai au 30 Août 2024

Tuteur universitaire : Véronique THELEN

Université Toulouse 3 Paul Sabatier
118 route de Narbonne
31062 Toulouse

Nom et qualité du Maître de stage :
LUMET Eliott, Post-Doctorant

Méthodes de réduction de dimension par apprentissage pour l'émulation statistique d'un modèle de dispersion atmosphérique

Emmarius Delar

(delaremmarius@gmail.com)

Du 2 Mai au 30 Août 2024

Remerciements

Je profite de ce rapport, qui clôture mes quatre mois de stage, pour remercier tous ceux qui ont contribué à rendre cette expérience inoubliable.

Avant tout, je souhaite remercier Eliott Lumet, Mélanie Rochoux et Etienne Gondet, mes encadrants, pour m'avoir permis de réaliser ce stage, qui a su renforcer mon intérêt pour le monde de la recherche.

Il me tient également à cœur de remercier tout particulièrement Eliott, pour qui j'ai développé une grande admiration. Par sa pédagogie, sa méthode d'encadrement et sa qualité de partage, notamment en partageant son attrait pour la recherche, il m'a d'autant plus motivé à m'investir pleinement dans mon stage. Je suis vraiment reconnaissant du privilège de l'avoir eu comme encadrant et lui souhaite le meilleur pour la suite.

Je remercie l'ensemble de l'équipe GLOBC avec qui j'ai tant appris lors des sessions FIDLE avec Anne, Delphine, Sophie, Luciano, Clara, Fernando, Saloua et Eliott, ainsi que lors de discussions sur la deuxième année de Master, les stages ou la poursuite en thèse avec Lilian, Romain, Théo, William, Quentin, Hugo, Camille, Maxime, Juliette et François. Je leur suis surtout reconnaissant pour les efforts qu'ils ont faits pour m'intégrer, malgré le fait que je sois une personne plutôt réservée.

Je tiens également à remercier ma collègue de bureau, Laure, pour avoir veillé à ce que je ne manque de rien durant mon stage.

Un grand merci à Mélanie, pour tous ces précieux conseils débouchant sur de nouvelles pistes à explorer, ces longues discussions sur les thèses et la vie de chercheur, mais surtout pour l'intérêt qu'elle a porté à mes travaux et à ma personne.

À Etienne de m'avoir permis d'avoir accès à un panel de formations CNRS et pour son dévouement dans l'organisation de la journée des stages.

Sachez que je vous suis extrêmement reconnaissant de m'avoir permis de vivre cette expérience unique.

DELAR Emmarius

Préambule

Dans le cadre de ma formation bi-diplômante en Master 1 de Mathématiques Appliquées et Statistiques ainsi qu'en Magistère 2 de Statistique et Modélisation Économique à l'Université de Rennes, j'ai eu l'occasion d'effectuer un stage de recherche au Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS) à Toulouse. Ce stage a été financé par l'Observatoire Midi-Pyrénées (OMP), une fédération de laboratoires des sciences de l'univers et de l'environnement de l'Université Toulouse III Paul Sabatier, via l'initiative ENV'IA qui vise à promouvoir et développer la synergie entre l'intelligence artificielle et les disciplines fondamentales des laboratoires de l'OMP.

Le CERFACS, qui est membre de l'OMP via l'unité mixte de recherche Climat, Environnement, Couplages et Incertitudes (CECI), est un laboratoire spécialisé dans le calcul de haute performance (HPC) et la conceptualisation d'outils de modélisation et de simulation. Disposant d'un savoir faire reconnu dans le monde de la recherche mais aussi dans l'industrie, le CERFACS a un statut atypique puisqu'il s'agit d'un centre de recherche privé fondé autour de 7 partenaires : Airbus, le CNES, EDF, Météo France, l'ONERA, Safran et TotalEnergies. Il s'articule autour de quatre équipes :

- Informatique et Support Utilisateur (CSG) : chargée de la gestion du parc informatique du laboratoire.
- Simulation numérique des écoulements de fluides (CFD) : focalisée sur le développement de méthodes numériques avancées pour la simulation d'écoulements. Elle tisse de nombreuses collaborations avec l'industrie, notamment dans le secteur de l'aéronautique.
- Algorithmes Parallèles & sCientific sOftware Operational Perfomances (ALGO-COOP) : chargée de faire le lien entre les besoins de la recherche fondamentale et la mise en application sur des cas réels.
- Modélisation du climat et de son changement global (GLOBC) : qui étudie le climat, son changement, et les risques environnementaux associés comme les extrêmes climatiques, les crues, les feux de forêt, et la pollution de l'air.

Ce stage, dont j'exposerai le déroulé au fil de ce rapport, a été réalisé au sein de l'équipe GLOBC. Le sujet de l'étude est la dispersion atmosphérique de polluants en milieu bâti. Il a pour finalité la construction de modèles réduits de mécanique des fluides par apprentissage. Ce stage se situe ainsi à l'interface entre mathématiques appliquées et physique et au carrefour des différents axes de recherche du CERFACS.

Mots clés : Dispersion de polluants · Modèle réduit · Réduction de dimension · Apprentissage automatique · Apprentissage profond · Auto-encodeur.

Table des matières

Introduction **4**

1 Émulation statistique d'un modèles de dispersion atmosphérique **6**
1.1 Campagne expérimentale MUST 6
1.2 Le modèle LES de l'expérience MUST #1682829 6
1.3 La réduction du modèle LES par apprentissage statistique 8
1.3.1 Principe 8
1.3.2 La base de données d'apprentissage de prévisions LES 9
1.4 Le modèle réduit POD–GPR 10
1.4.1 Principe 10
1.4.2 l'Analyse en Composantes Principales (ACP) 11
1.4.3 Régression des coefficients latents par processus gaussiens 11
1.4.4 Application à l'émulation du modèle LES du cas MUST 11
1.5 Positionnement du stage 13

2 État de l'art des méthodes de réduction de dimension **15**
2.1 Méthodes d'apprentissage automatique 15
2.1.1 L'analyse en composantes principales à noyau (KPCA) 16
2.1.2 La méthode des moindres carrés partiels (PLS) 18
2.2 Méthodes d'apprentissage profond 18
2.2.1 Les auto-encodeurs convolutifs (CAE) 19
2.2.2 Quelques auto-encodeurs pour les graphes 20

3 Bilan des résultats obtenus **23**
3.1 Démarche de comparaison 23
3.1.1 Simplifications 23
3.1.2 Erreur de réduction-reconstruction 23
3.1.3 Les métriques de validation 23
3.2 Calibration des méthodes de réduction 24
3.2.1 Approche PLS 24
3.2.2 Approche ACP à noyau (KPCA) 24
3.2.3 Approche auto-encodeurs convolutifs 26
3.3 Inter-comparaisons des modèles 27
3.4 Apprentissage par transfert 30

Bilan du stage **31**

Introduction

Contexte Au cours de ces dernières années, de nombreux accidents industriels ont conduit au rejet d'importantes quantités de polluants dans l'atmosphère. L'incendie de l'usine de produits chimiques de la société Lubrizol à Rouen en Septembre 2019 (**Fig.** 1) en est un parfait exemple. La vulnérabilité des populations avoisinantes face au risque d'intoxication représente un enjeu sanitaire de grande ampleur. Il est donc nécessaire pour les organismes de santé public, tels que l'ARS, de prédire les différences scénarios de dispersion de polluants afin de prévenir, anticiper et gérer au mieux ces évènement. Dans ce contexte, les scientifiques et ingénieurs jouent un rôle clé par le développement de modèles permettant de reproduire la dispersion de polluant en milieu urbain.

Figure 1 – *Photo du panache de polluants émis lors de l'incendie de l'usine Lubrizol à Rouen en 2019. Crédit photo : Jean Pierre Mauger.*

La dispersion des polluants en milieu urbain est un problème complexe de dynamique des fluides. Cela nécessite de représenter le plus fidèlement possible la couche limite atmosphérique, la source d'émission des polluants, ainsi que l'interaction du milieu bâti avec l'écoulement atmosphérique. À ce jour, la littérature montre que la capacité des modèles de mécanique des fluides numérique (CFD en anglais), qui résolvent explicitement les équations de l'écoulement et de la dispersion, à prendre en compte l'effet des bâtiments de la canopée urbaine sur l'écoulement atmosphérique leur confère un précision accrue par rapport aux autres approches de modélisation de la dispersion [Holmes and Morawska, 2006, Blocken et al., 2013].

La fidélité des modèles de CFD dépend principalement de la méthode utilisée pour la représentation de la turbulence et du degré de discrétisation du domaine d'étude. Utiliser un maillage de l'espace fin pour la résolution des équations gouvernantes permet de mieux prendre en compte les plus petite échelles de turbulence ce qui augmente la précision du modèle, en contrepartie d'un coût de calcul très important. En effet, effectuer une prévision de dispersion à micro-échelle (i.e. à l'échelle d'un quartier) avec un modèle CFD coûte typiquement entre 10^2 et 10^5 heure de calcul cœur. Ce temps de calcul très important limite significativement l'utilisation des modèles CFD en conditions opérationnelles, qui nécessitent d'être réactif en cas d'accident et d'effectuer un grand nombre de simulations pour prendre en compte chaque scénario de dispersion possible.

Pour répondre à ce problème de coût computationnel, une solution consiste à construire par apprentissage automatique des modèles réduits. Ces modèles apprennent les relations entre les entrées et les sorties du modèle physique complexe à partir d'un ensemble de réalisations de ce modèle. Une fois entraîné, un modèle réduit est capable d'inférer les résultats qu'aurait fourni le modèle physique haute-fidélité en quelques secondes seulement.

Positionnement du stage Au cours de mon stage, je me suis intéressé à la construction d'un modèle réduit pour émuler un modèle CFD dit de simulation aux grandes échelles de turbulence (LES en anglais, pour *Large-Eddy Simulation*) qui reproduit la campagne expérimentale de dispersion MUST [Biltoft, 2001, Yee and Biltoft, 2004]. Ce travail s'inscrit dans la continuité des thèses de Nony [2023] et Lumet [2024] qui ont montré qu'une approche de réduction de modèle appelée POD–GPR [Marrel et al., 2015] permet d'obtenir des résultats pertinents pour l'émulation de la concentration moyenne en fonction de paramètres de condition aux limites météorologiques (la direction du vent et sa vitesse), en plus de fournir une estimation de l'incertitude associée aux prévisions de dispersion. Le modèle POD–GPR est basé sur deux étapes : i) la réduction de la dimension des champs d'intérêt en concentrant l'information dans un espace de représentation, l'*espace latent*, avec une approche d'analyse en composantes principales (ACP, plus communément appelé POD en CFD [Sirovich, 1987, Berkooz et al., 1993]), ii) l'apprentissage du lien entre les paramètres d'entrée du modèle LES et l'espace latent qui se fait à l'aide de processus gaussiens de régression (GPR en anglais).

L'analyse des résultats de l'approche POD–GPR montre qu'elle permet d'atteindre des niveaux de précision satisfaisants dans l'estimation du champ de concentration moyen, en plus de fournir une estimation correcte de l'incertitude associée [Lumet, 2024]. Toutefois ces résultats restent améliorables, notamment dans les zones de fortes concentrations proches de la source. Il a été montré que les erreurs commises par le modèle POD–GPR sont en grande partie dues à l'erreur faite lors de l'étape de réduction de dimension par la POD [Lumet, 2024]. La POD étant une méthode linéaire, nous pouvons faire l'hypothèse que les limites du modèle réduit de référence proviennent de la non-linéarité de la réponse du champ de concentration moyen aux variations des paramètres d'entrée, et notamment la direction du vent.

Problématique L'objectif de mon stage est de trouver des méthodes d'intelligence artificielle pouvant capter les non-linéarités dans la base d'apprentissage de LES dans le but d'améliorer l'étape de réduction de dimension du modèle réduit.

Déroulé du stage Pour répondre à la problématique de mon stage, j'ai tout d'abord étudié et pris en main les outils existants, développés au CERFACS dans le cadre des thèses de Nony [2023] et Lumet [2024] (Section [1]). J'ai par la suite mené une revue de l'état de l'art afin d'identifier les nouvelles méthodes de réduction de dimension les plus pertinentes pour mon stage, que je présente dans la Section [2]. Pour finir, j'ai implémenté et optimisé les méthodes sélectionnées, avant de les analyser et de les comparer avec la méthode de référence, ce qui fait l'objet de la Section [3].

1 Émulation statistique d'un modèles de dispersion atmosphérique

Mon stage s'inscrit dans la suite des thèses de Nony [2023] et Lumet [2024] sur les limites des modèles de CFD des processus atmosphériques à micro-échelle. Ces travaux, réalisés au CERFACS, ont permis d'évaluer les incertitudes d'un modèle de CFD de simulation aux grandes échelles de turbulence (LES) dans le cadre de la reproduction d'une expérience terrain : la campagne MUST [Biltoft, 2001, Yee and Biltoft, 2004].

Effectuer une simulation LES demande un temps de calcul considérable en raison de sa très fine résolution spatio-temporelle. Par conséquent, la quantification des incertitudes du modèle LES est limitée par le nombre d'évaluations réalisables en un temps raisonnable. Pour surpasser ce problème, un modèle réduit de type POD–GPR [Marrel et al., 2015] est utilisé comme émulateur de la surface de réponse du modèle LES pour différentes conditions météorologiques.

Ces outils ayant déjà été développés par le CERFACS, le début de mon stage a consisté à les comprendre puis les prendre en main. Cette phase m'a permis de recenser les limites du modèle POD–GPR et de définir ma problématique.

1.1 Campagne expérimentale MUST

La campagne MUST vient de l'initiative de chercheurs conscient des limites des expériences de laboratoire dans la caractérisation des processus de dispersion en milieu urbain [Biltoft, 2001, Yee and Biltoft, 2004]. Elle a été réalisée sur un site de l'armée américaine, Dugway Proving Ground, au nord-ouest du désert de l'Utah, qui sert de zone de test pour les systèmes de défenses face à des attaques chimique ou biologique. Sur une parcelle de 200x200 m^2 de cette zone, 10 lignes de 12 conteneurs de 12.2m de long, 2.42m de large, 2.54 de haut ont été placés pour représenter une canopée urbaine simplifiée (**Fig.** 2a). Un gaz, le propène, a ensuite été relâché au cours de plusieurs essais à différentes positions et pour différentes conditions atmosphériques. Dans cette étude nous nous concentrons sur l'essai #2681829 sous conditions atmosphériques neutres.

Au cours de cette campagne, un grand nombre de capteurs ont été installés pour mesurer la vitesse du vent et la concentration en gaz (**Fig.** 2b). Ces mesures ont servi à valider le modèle LES présenté Section [1.2] mais n'ont pas été utilisées pour le problème d'apprentissage de mon stage.

Le cas de l'expérience MUST est intéressant car : i) la représentation d'une canopée urbaine idéalisée **Fig.** (2a) facilite la discrétisation du domaine sous forme de maillage pour la résolution numérique, ii) le fait que la zone d'intérêt soit située dans un désert isolé simplifie les hypothèses à faire sur les conditions météorologique, iii) le gaz utilisé lors de cette expérience, le propène, ne réagit pas chimiquement avec l'atmosphère à l'échelle de l'expérience.

1.2 Le modèle LES de l'expérience MUST #1682829

La modélisation en mécanique des fluides permet de caractériser les processus complexes que suivent les fluides à l'état gazeux ou liquide lors de leur écoulement en s'appuyant sur les équations de Navier-Stockes. En pratique, il est nécessaire d'avoir recours à des méthodes de simulation numérique pour résoudre ces équations. La simulation aux grandes échelles de turbulence (LES pour Large Eddy Simulation en anglais) est une approche de CFD qui se situe entre l'approche RANS (Moyenne de Reynolds) qui résout les équations de Navier-Stockes en les moyennant et l'approche DNS (Simulation Numérique Direct) qui les résout explicitement. Avec l'approche LES, les équations sont filtrées et les grandes échelles (i.e. plus grandes que la

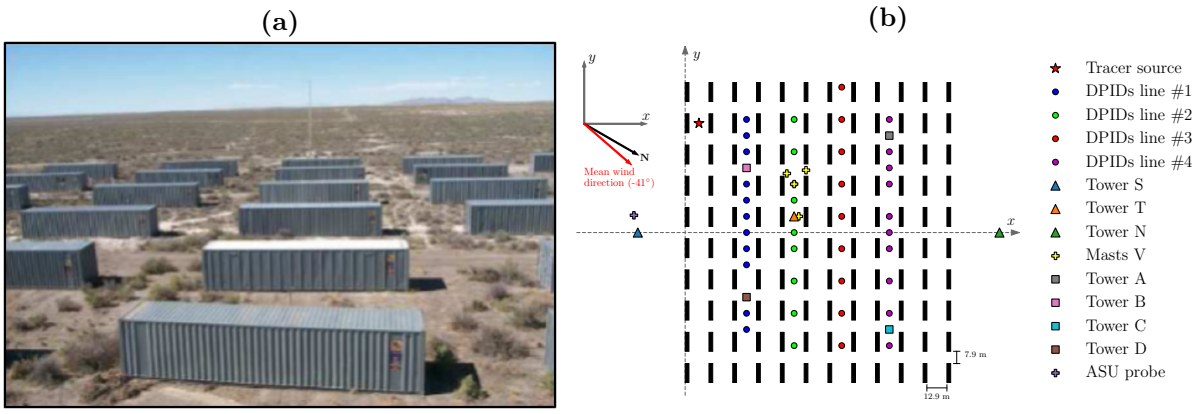


Figure 2 – *(a) Photo du site où s'est déroulé l'expérience MUST par Schatzmann et al. [2010] ; (b) Représentation schématique de la configuration de la zone de test pour la campagne MUST par Lumet [2024].*

taille du maillage) sont résolues explicitement, tandis que l'effet des structures turbulentes de plus petites échelles sont modélisées.

La LES a ainsi pour avantage d'avoir un temps de calculs moins important que la DNS tout en réduisant l'incertitude liée à la modélisation de la turbulence comparé à l'approche RANS (Gousseau et al. [2011], García-Sánchez et al. [2018]). De plus, elle donne la possibilité de quantifier l'incertitude aléatoire des grandeurs d'intérêt induite par la variabilité interne de la couche limite atmosphérique [Lumet et al., 2024].

Dans la suite je m'intéresse au modèle LES construit par Rochoux et al. [2021] et Lumet et al. [2024] pour reproduire l'expérience MUST #1682829. Ce modèle est basé sur le code AVBP [1] développé au CERFACS pour effectuer des LES et DNS pour des applications industrielles (combustion, écoulements aérodynamiques, dispersion de polluants). Ce code permet de résoudre en parallèle, sur des architectures CPU et GPU, les équations de Navier-Stokes pour différents types de maillages : structurés, non structurés et hybrides. La discrétisation du domaine sur lequel à été réalisé la reproduction LES du cas MUST s'est faite avec un maillage non structuré de $N_{nodes} \approx 90 \times 10^6$ nœuds (**Fig.** 3).

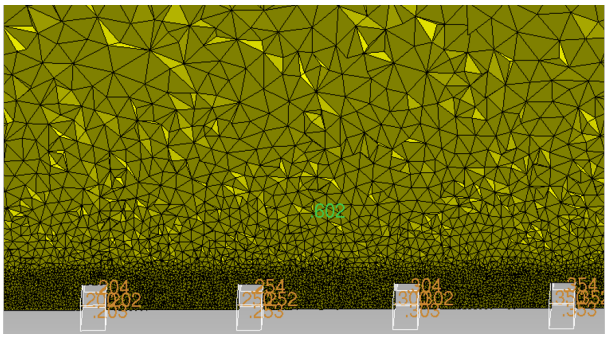


Figure 3 – *Coupe verticale au niveau des conteneurs du maillage du modèle du cas MUST à partir duquel les équations de la LES sont discrétisées.*

Simuler la dispersion de polluants avec le solveur AVBP pour des conditions limites fixées permet d'obtenir, après 20 000 heures CPU, une reproduction du phénomène sur 200 secondes à

1. `https://cerfacs.fr/avbp7x/`

partir du moment où le gaz est relâché (**Fig.** 4). Ce coût de calcul de cette méthode de simulation est trop élevé pour effectuer i) une simulation dans un cas d'urgence ou ii) un grand ensemble de simulations pour améliorer la quantification des incertitudes intrinsèque du modèle.

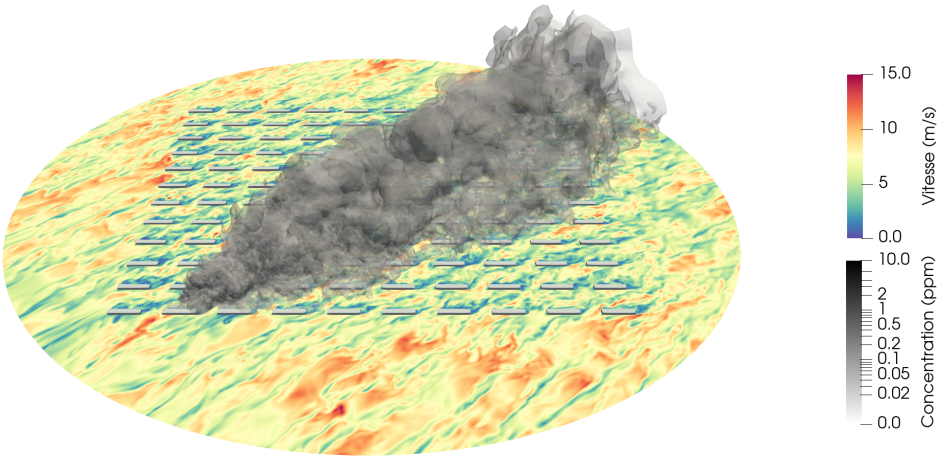


Figure 4 – *Simulation aux grandes échelles de tourbillons de l'écoulement atmosphérique et de la dispersion d'un polluant dans une canopée urbaine simplifiée par [Lumet, 2024]*

Pour répondre à ce problème, un ensemble de simulations ont été réalisées dans le but de constituer une base de données d'apprentissage pour construire un modèle réduit pour fournir des prévisions de dispersion en un temps réduit.

1.3 La réduction du modèle LES par apprentissage statistique

1.3.1 Principe

Un modèle réduit peut être défini comme une version simplifié d'un modèle haute-fidélité, souvent coûteux (ici le modèle LES présenté dans la Section 1.2). Le principe de la réduction de modèle, illustré **Fig.** 5, consiste à apprendre la surface de réponse du modèle à émuler à partir d'un ensemble de réalisations de ce dernier afin d'inférer de nouveau résultats uniquement à partir d'informations mises en entrée, ici (α_{inlet}, u_*).

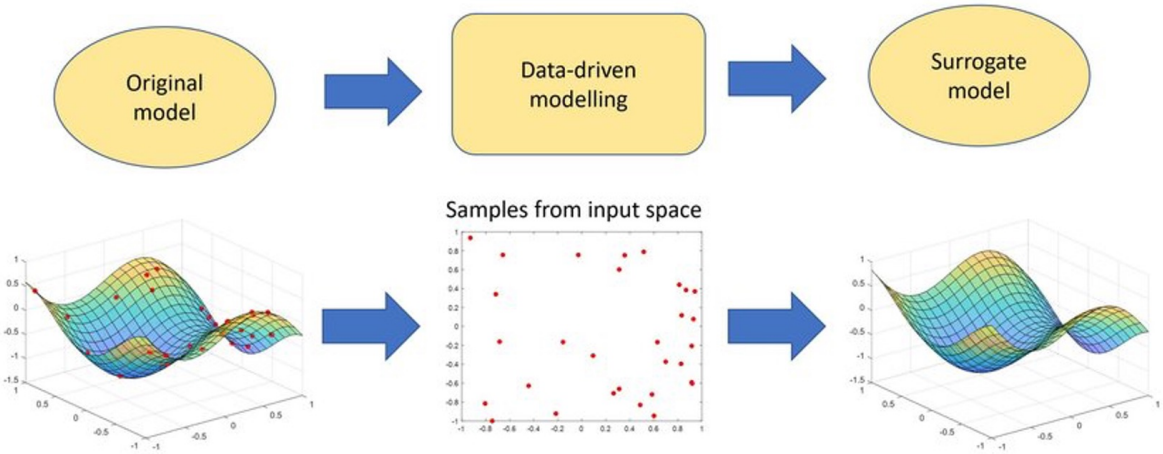


Figure 5 – *Le principe du modèle réduit par [Kocijan et al., 2023]*

À partir de la base de données de prévisions LES, Lumet [2024] a entraîné et validé un premier modèle réduit en suivant une approche POD–GPR Marrel et al. [2015]. Les prochaines sous-sections introduisent la base de données utilisée et le principe de cette approche et présentent les résultats obtenus pour l'émulation du modèle LES du cas MUST.

1.3.2 La base de données d'apprentissage de prévisions LES

La base de données construite par Lumet [2024], à partir de laquelle j'ai testé les différentes méthodes d'apprentissage au cours de mon stage, contient un ensemble de 200 réalisations générées à partir du modèle LES. Cette base de données explore la réponse du modèle LES au forçage atmosphérique méso-échelle qui est représenté par deux paramètres : la direction du vent $\alpha_{inlet}(°)$ et la vitesse de friction $u_*(\mathrm{ms}^{-1})$.

Le couple de ces deux paramètres forme les entrées du modèle LES, définies sur l'espace :

$$(\alpha_{inlet}, u_*) \in [-90°, 30°] \times [0.07\,\mathrm{m\,s^{-1}}, 0.89\,\mathrm{m\,s^{-1}}].$$

qui a été défini à partir d'une micro-climatologie afin d'être représentatif des conditions météorologiques usuelles du terrain d'expérimentation de la campagne MUST.

Cet espace des paramètres a ensuite été échantillonné pour former 200 couples de paramètres à l'aide de la séquence de Halton. La séquence de Halton est une méthode déterministe qui permet d'échantillonner la surface de réponse du modèle LES de manière à la couvrir uniformément, ce qui évite de privilégier de fraction de l'espace d'entrée du modèle en dépit d'une autre lors de l'apprentissage. Il s'agit d'une séquence à faible discrépance, ce qui signifie qu'elle évite d'échantillonner plusieurs fois la même zone comparé à un échantillonnage aléatoire (**Fig.** 6), ce qui permet de maximiser l'information apporté par chaque nouvel échantillon.

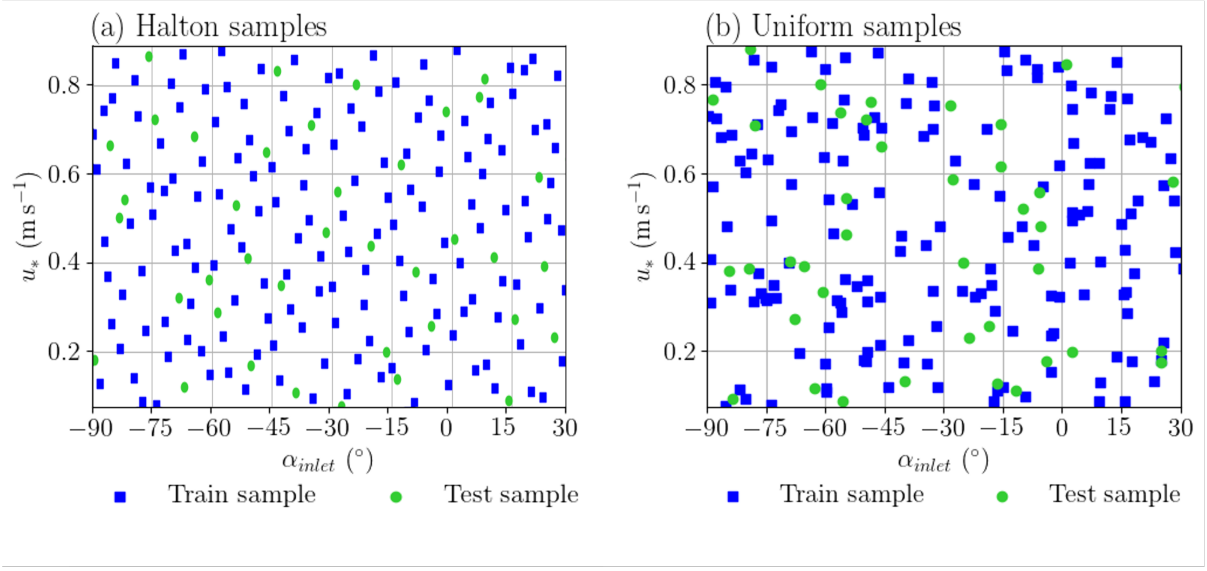


Figure 6 – *Génération de 200 points dans l'espace des paramètres avec une séquence de Halton (a) et (b) génération de 200 points dans l'espace des paramètres avec un tirage aléatoire uniforme (b).*

Le modèle LES est ensuite évalué pour fournir les 200 prévisions de la concentration moyenne en propène à chaque jeu de conditions aux limites. Cette base de données est ensuite classiquement divisée en une base de 160 échantillons dédiée à l'entraînement des modèles réduits et une base de 40 échantillons de test dédiée à leur évaluation (**Fig.** 6). Au total, le coût de cet ensemble de simulations est de 5.7 millions d'heures CPU.

1.4 Le modèle réduit POD–GPR

1.4.1 Principe

L'approche POD–GPR est aujourd'hui une approche standard [Vinuesa et al., 2020] qui a déjà montré de bons résultats pour l'émulation de d'écoulements et de dispersion en milieu urbain à micro-échelle [Xiao et al., 2019, Xiang et al., 2021, Nony et al., 2023]. Le principe fondamental de cette approche est de combiner :

i) une *étape de réduction* qui consiste à trouver un espace de faible dimension, appelé *espace latent*, de dimension $L \ll N_{nodes}$, sur lequel sont projetés les champs à émuler $\mathbf{y}$ (ici le champ de concentration moyenne), et qui est réalisée à l'aide d'une méthode appelée *proper orthogonal decomposition* (POD) [Sirovich, 1987, Berkooz et al., 1993], équivalente à l'analyse en composantes principales (ACP) ;

ii) et une *étape de régression* utilisant la régression par processus gaussien (GPR) [Rasmussen et al., 2006], qui consiste à apprendre, à partir de l'ensemble d'entraînement, la relation entre les paramètres d'entrée $\boldsymbol{\theta}$, ici les paramètres de conditions aux limites (α_{inlet}, u_*), et les coefficients de la projection des champs sur l'espace latent $\{k_\ell(\boldsymbol{\theta})\}_{\ell=1}^{L}$.

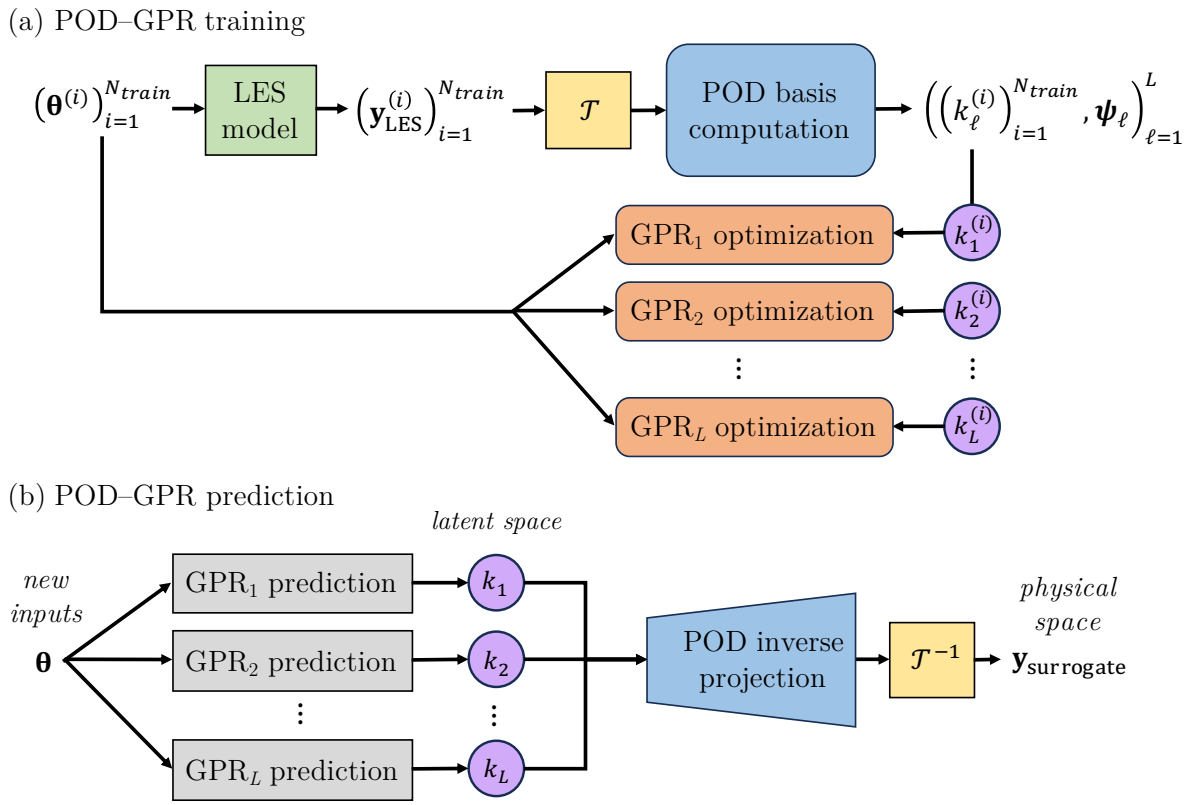


Figure 7 – *Schéma de principe du modèle réduit POD–GPR par Lumet [2024].*

Cette approche de réduction-régression permet i) de réduire la dimension du problème de régression à L variables latentes ($L \ll N_{nodes}$) et donc de réduire considérablement la charge de calcul de la tâche d'apprentissage ; et ii) de séparer la dépendance paramétrique du champ de la variabilité spatiale (c.f. Eq. 1).

Le modèle POD–GPR est mis en œuvre selon une approche d'apprentissage statistique standard, c'est-à-dire avec une phase d'entraînement initiale consistant à i) prétraiter les champs LES (opérateur dénoté $\mathcal{T}$), ii) construire la base réduite POD à partir de la base d'échantillons

d'entraînement, et iii) l'optimisation des modèles GPR dans l'espace latent (**Fig.** 7a). Une fois entraîné, le modèle POD–GPR peut fournir des prédictions de champs pour de nouvelles entrées $\boldsymbol{\theta}$ de la manière suivante : i) les coefficients réduits POD associés sont prédits par les modèles GPR, et ii) la projection POD inverse et le pre-traitement inverse sont appliquées à ces coefficients pour récupérer le champ physique (**Fig.** 7b). Les sections suivantes présentent les éléments théoriques des techniques POD et GPR nécessaires à cette étude.

1.4.2 l'Analyse en Composantes Principales (ACP)

L'ACP (ou POD) est une méthode d'apprentissage non supervisée pour la réduction de dimension [Saporta, 2006]. Avec cette technique, les champs d'intérêt $\mathbf{y}$ sont projetés sur la base des vecteurs propres $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{N_{train}})$ de la matrice de variance-covariance des données d'apprentissage. La base des vecteurs propres est ensuite ordonnée selon la part de la variance totale des données d'entraînement que chaque axe parvient à expliquer, et tronquée pour conserver uniquement les L premiers vecteurs.

La décomposition d'un champ $\mathbf{y}$ sur l'espace latent s'écrit alors :

$$\mathcal{T}(\mathbf{y}) = \sum_{\ell=1}^{L} k_\ell \boldsymbol{\phi}_\ell. \tag{1}$$

avec $\mathcal{T}$ l'éventuel pré-traitement appliqué aux données avant de construire la base de projection $(\boldsymbol{\psi}_\ell)_{\ell=1}^{L}$, et k_ℓ les coefficients latents associés à la projection.

L'ACP possède des propriétés pertinentes pour notre application tels que :

— la hiérarchisation des axes par quantité de variance portée, qui permet de filtrer le bruit dans les données d'apprentissage

— le fait qu'il y ait qu'un unique hyperparamètre à définir (la dimension de l'espace latent)

Cependant, le principal défaut de cette méthode réside dans le fait qu'elle ne soit efficace qu'avec des variables corrélées linéairement [Gruber et al., 2022a, Csala et al., 2022, Murata et al., 2020]. Il serait donc intéressant de pouvoir généraliser cette méthode à des cas non-linéaire pour répondre à notre problématique.

1.4.3 Régression des coefficients latents par processus gaussiens

Pour l'étape de régression, L processus gaussiens (GPs) indépendants sont construits pour apprendre la relation les paramètres d'entrée $\boldsymbol{\theta}$ et les L coefficients latents $\{k_\ell(\boldsymbol{\theta})\}_{\ell=1}^{L}$ associés au champ d'intérêt $\mathbf{y}(\boldsymbol{\theta})$ (**Fig.** 7). Il s'agit d'une méthode d'apprentissage bayésien [Rasmussen et al., 2006] où la distribution des coefficients associés $\{k_\ell(\boldsymbol{\theta}^*)\}_{\ell=1}^{L}$ à de nouvelles données d'entrée $\boldsymbol{\theta}^*$ est inférée à partir de la base d'apprentissage $\left(\boldsymbol{\theta}^{(i)}, k_\ell(\boldsymbol{\theta}^{(i)})_{\ell=1}^{L}\right)_{i=1}^{N_{train}}$.

En plus de leur robustesse et de leur simplicité, un des avantages significatifs des GPs est qu'il fournisse une estimation de l'erreur de régression, dont on se sert pour quantifier l'incertitude totale du modèle POD–GPR [Lumet, 2024].

1.4.4 Application à l'émulation du modèle LES du cas MUST

Dans cette section, je résume brièvement comment Lumet et al. [2024] a appliqué l'approche POD–GPR à l'émulation du modèle LES du cas MUST présenté Sect. [1.2]. Ce modèle réduit constitue la référence que mon stage vise à améliorer.

Pré-traitement La première étape du pré-traitement consister à interpoler les champs sur un maillage d'analyse avec un nombre de nœuds réduits d'un facteur 10 ($N_{nodes} \approx 1.8 \times 10^6$) pour réduire la dimension du problème. Les données d'entraînement sont ensuite centrées comme il convient de le faire pour l'ACP [Berkooz et al., 1993]. Enfin, Lumet [2024] a constaté qu'appliquer une transformation logarithmique aux données d'apprentissage permettait d'améliorer significativement les performances du modèle réduit POD–GPR en réduisant les différences d'échelle entre les points de la base. Ces différences d'échelle sont dues au fait que la concentration varie sur plusieurs ordre de grandeurs ; de 10^2 ppm proche de la source, elle décroît exponentiellement jusqu'à 10^-4 ppm. Le pré-traitement des données est ainsi défini comme suit :

$$\begin{aligned} \mathcal{T}_{log} : \mathbb{R}^{N_{nodes}} &\longrightarrow \mathbb{R}^{N_{nodes}}, \\ \mathbf{y}(x_k) &\longmapsto \sqrt{\frac{\omega(x_k)}{\Omega}} \left[ln(\mathbf{y}(x_k) + y_t) - \langle \ln(\mathbf{y}_{\mathrm{LES}}(x_k) + y_t) \rangle \right], \ 1 \leq k \leq N, \end{aligned} \tag{2}$$

avec $\mathbf{y}(x_k)$ le champ k, $\langle \mathbf{y}(x_k) \rangle$ le champ moyen de l'ensemble d'entraînement, $\omega(x_k)$ le volume du nœud k, Ω le volume total du domaine, et y_t un seuil pour éviter les problèmes avec les concentrations proches de zéro. Sa transformation inverse s'écrit :

$$\begin{aligned} \mathcal{T}_{\log}^{-1} : \mathbb{R}^N &\longrightarrow \mathbb{R}^N, \\ \widetilde{\mathbf{y}}(x_k) &\longmapsto \exp\left(\sqrt{\frac{\Omega}{\omega(x_k)}} \widetilde{\mathbf{y}}(x_k) + \langle \ln(\mathbf{y}_{\mathrm{LES}}(x_k) + y_t) \rangle \right) - y_t, \end{aligned} \tag{3}$$

Calibration du POD–GPR Concernant l'étape de réduction de dimension, seules les 10 premières composantes de l'ACP sont conservées. On projette ainsi des champs de concentration 3-D de dimension $N_{node} \approx 90 \times 10^6$ dans un espace latent de dimension $L = 10$ seulement. En effet, Lumet [2024] montre que c'est suffisant pour minimiser l'erreur de réduction. De plus, comme illustré (**Fig.** 8), utiliser d'avantage de modes introduit du bruit dans les surfaces de réponse du modèle POD–GPR [Lumet, 2024].

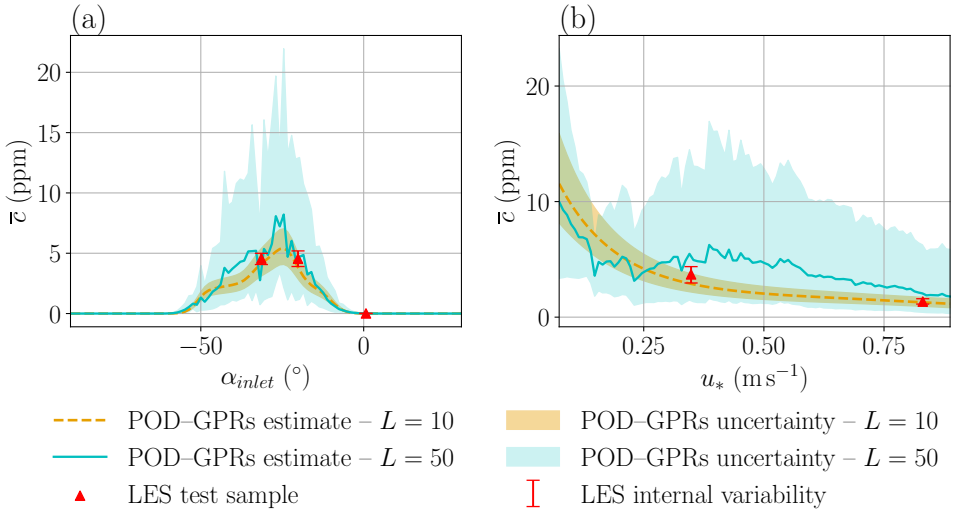


Figure 8 – *Quantification de l'incertitude liée au prévision du modèle POD–GPRR pour 10 composantes et pour 50 composantes par [Lumet, 2024]*

Concernant les procesus gaussiens, leurs hyper-paramètres sont optimisés par maximisation de la log-vraissemblance [Hastie et al., 2009]. Pour plus de détails, le lecteur est renvoyé à Lumet [2024].

Résultats Sur ce cas d'étude, l'entraînement du modèle POD–GPR dure moins de 15 secondes, et il suffit de 8 millisecondes pour inférer une nouvelle prévision du champ de concentration associé à un nouveau jeux de paramètres de conditions météorologiques (α_{inlet}, u_*). Cela représente un gain de temps drastique par rapport au modèle LES qui fournit une prévision en 20 000 heures, le POD–GPR répond donc bien au besoin pour lequel il a été conçu.

Comme illustré (**Fig.** 9), pour un échantillon de test, le modèle POD–GPR parvient bien à reproduire précisément la structure du champ de concentration entre les obstacles, comme illustré par les isos de concentration 0.01ppm et 5ppm (**Fig.** 9d). On note toutefois une surestimation au niveau de la source d'émission (**Fig.** 9c). En moyenne, on trouve que le modèle POD–GPR reproduit les champs LES de l'échantillon de test avec un niveau de précision quasiment maximal compte tenu de l'incertitude de ces champs [Lumet, 2024]. Il reste néanmoins un certain nombre d'aspects pour lesquels les performances du POD–GPR peuvent être améliorées. Enfin, on note que le modèle POD–GPR fournit une estimation cohérente de l'incertitude associée à ses prédiction (**Fig.** 8).

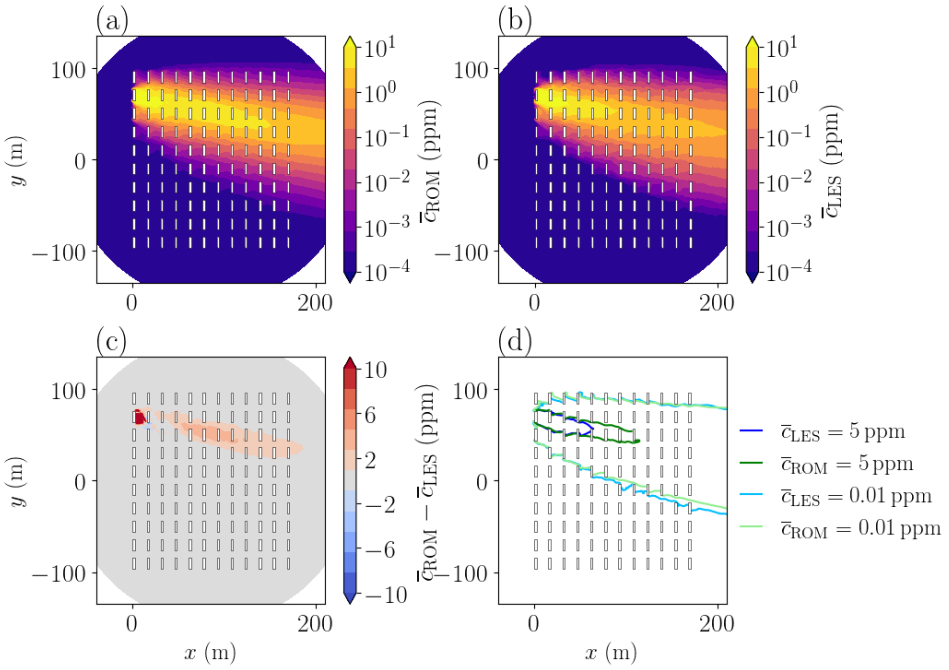


Figure 9 – *Coupe horizontal à 1.6m de hauteur du champ de concentration évalué pour des paramètres* $(\alpha_{inlet}, u_*) = (-8.9°, 0.38\,m\,s^{-1})$ *par le modèle LES (a), par le modèle réduit POD–GPR (b). (c) Coupe horizontale de la différence absolue entre les deux prévisions, et (d) comparaison des iso-concentrations 5-ppm et 0.01-ppm.*

1.5 Positionnement du stage

Bien que performant, le modèle réduit POD–GPR commet toujours des erreurs significatives en reproduisant les prévisions de concentration du modèle LES, en particulier dans les zones de fortes concentration [Lumet, 2024] surtout au niveau de la source d'émission (**Fig.** 9).(c).

Comme je l'illustre (**Fig.** 10) pour deux métriques introduites Section [3.1.3], ces erreurs sont principalement liées à l'étape de réduction de dimension. L'ACP étant une méthode de réduction de dimension linéaire, elle n'est pas adaptée pour représenter la relation non-linéaire entre les champs de concentration et les paramètres d'entrée (α_{inlet}, u_*), non-linéarité notamment induite par l'effet des conteneurs sur l'écoulement. De plus, augmenter la dimension de l'espace latent

ne corrige pas le problème mais détériore les prévisions du modèle POD–GPR (**Fig.** 8).

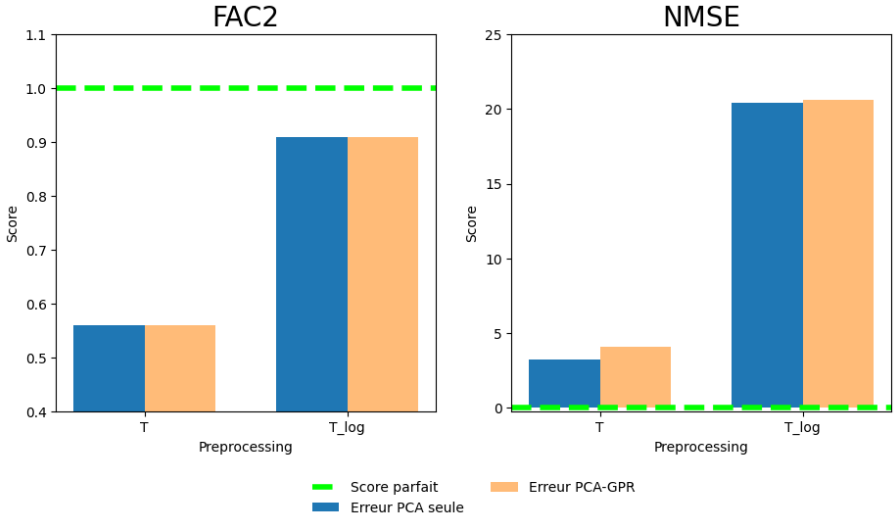


Figure 10 – *Quantification des erreurs FAC2 à gauche et NMSE à droite (voir Section [3.1.3]) commises par l'étape de projection inverse de la POD pour l'espace latent en bleu et des erreurs commises par le modèle POD-GPR en orange. Où $\mathcal{T}_{log}$ se réfère à* ***Eq.*** *2) et $\mathcal{T}$ représente sa version sans transformation log.*

L'objectif de mon stage est d'explorer le potentiel de méthodes d'apprentissage statistique innovantes pour améliorer l'étape de réduction de dimension du modèle réduit. L'enjeu sous-jacent est donc de trouver une méthode qui parvient mieux à capter les non-linéarités dans la base d'apprentissage de LES tout en continuant à compresser au mieux les données dans un espace latent de faible dimension.

2 État de l'art des méthodes de réduction de dimension

Dans cette section je présente l'état de l'art que j'ai réalisé afin de trouver des méthodes de réduction de dimension pour remplacer l'ACP. L'état de l'art ou la revue de la littérature est l'étape préliminaire à un travail de recherche. Elle consiste à recenser les méthodes actuelles utilisées dans le domaine que l'on souhaite étudier.

Pour mon stage j'ai donc passé en revue un ensemble d'article sur la réduction de dimension pour les émulateurs de modèle de CFD, dont ce que mes encadrants m'ont partagés dès le début de mon stage [Gruber et al., 2022b, Nony et al., 2023]. Cependant, la plus grande parti de mon état de l'art c'est fait à l'aide des outils mis à disposition par les deux universités partenaires de ma formation, l'Université de Rennes et l'Université Rennes 2, avec l'accès aux bibliothèques universitaires en ligne et les différents accès vers d'autres plateformes de partage de ressources scientifiques.

Je me suis également énormément appuyé sur les différents cours que j'ai suivi au tout au long de l'année universitaire notamment les cours d'apprentissage supervisé [1], d'apprentissage non-supervisé [Meynaoui, 2024], de méthodes de monte carlo par chaînes de Markov [Gavra, 2023], d'analyse de données [Cornillon, 2024] et d'économétrie avancée [Cadoret, 2024]

Les méthodes que j'ai trouvées peuvent être séparées en deux classes (**Fig.** 11) :

— les méthodes d'apprentissage automatique avec l'ACP standard, l'analyse en composantes principales par noyau (KPCA pour *kernel principal component analysis* en anglais) et la méthode des moindes carrés Partiels (PLS en anglais)

— les méthodes d'apprentissage profond basées sur les réseaux de neurones et notamment les architectures d'auto-encodeur convolutifs [Murata et al., 2020, Gruber et al., 2022b].

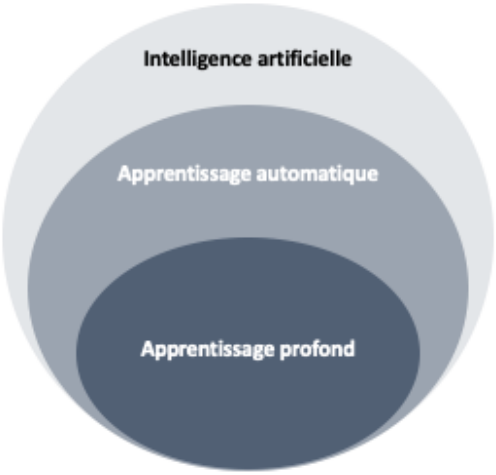


Figure 11 – *Schéma représentant l'intelligence artificielle et ces deux principaux sous-domaine*

2.1 Méthodes d'apprentissage automatique

L'apprentissage automatique (ou machine learning en anglais) est une sous branche de l'intelligence artificielle. Elle a pour objectif d'apprendre des phénomènes à partir d'information récoltées lors de leurs réalisations à l'aide de modèles et hypothèses mathématiques. Grâce au modèle d'apprentissage automatique, il est désormais possible d'automatiser des tâches difficiles comme la réduction de dimension.

1. `https://lrouviere.github.io/page_perso/cours/apprentissage_sup.html`

2.1.1 L'analyse en composantes principales à noyau (KPCA)

Pour palier la limitation de l'ACP à la représentation des données linéaires, l'astuce du noyau (ou kernel trick en anglais) a été introduite pour transformer des algorithmes linéaires basés sur un produit scalaire en algorithmes non-linéaires. Elle consiste à trouver un espace de représentation des données de dimension supérieur dans lequel elles peuvent être agrégées linéairement (**Fig.** 12). Basée sur le théorème de Mercer [1909] [1], cette astuce permet de calculer la matrice de variance-covariance de ce nouvel espace augmentée sans avoir besoin d'en expliciter la forme. Cela est possible grâce aux propriétés des espaces de Hilbert et des noyaux reproduisants [Paiement, 2004].

Soit $\mathbf{Y} \in \mathbb{R}^{N_{train} \times N_{nodes}}$, un jeu de N_{train} réalisations de champs de dimension N_{nodes}. On considère une application Φ quelconque qui permet de projeter les données dans l'espace de représentation augmenté pour en extraire de nouvelles caractéristiques :

$$\begin{aligned} \Phi : &\mathbb{R}^{N_{nodes}} \to \mathcal{H}^q \\ &\mathbf{y} \mapsto \Phi(\mathbf{y}) \end{aligned}$$

telle que :

$$\forall (x, x_0) \in X^2,\ f(x, x_0) = \langle \Phi(x), \Phi(x_0) \rangle_{\mathcal{H}} \tag{4}$$

avec $\mathcal{H}$ un espace de Hilbert de dimension arbitraire q, possiblement infinie, et f une fonction appelée noyau reproduisant [2] associé à Φ et qui coïncide avec le produit scalaire $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ dans $\mathcal{H}$

On souhaite maintenant appliquer l'ACP aux données projetées dans leur nouvel espace de représentation Z. On définit alors la matrice de variance-covariance de Z notée Σ_z :

$$\Sigma_z = \Phi(\mathbf{Y})^\top \Phi(\mathbf{Y}) = \begin{bmatrix} \Phi(\mathbf{y}_1)^\top \Phi(\mathbf{y}_1) & \Phi(\mathbf{y}_1)^\top \Phi(\mathbf{y}_2) & \cdots & \Phi(\mathbf{y}_1)^\top \Phi(\mathbf{y}_n) \\ \Phi(\mathbf{y}_2)^\top \Phi(\mathbf{y}_1) & \Phi(\mathbf{y}_2)^\top \Phi(\mathbf{y}_2) & \cdots & \Phi(\mathbf{y}_2)^\top \Phi(\mathbf{y}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(\mathbf{y}_n)^\top \Phi(\mathbf{y}_1) & \Phi(\mathbf{y}_n)^\top \Phi(\mathbf{y}_2) & \cdots & \Phi(\mathbf{y}_n)^\top \Phi(\mathbf{y}_n) \end{bmatrix},$$

or, d'après **Eq.** 4, il est possible d'utiliser le noyau f comme mesure de similarité pour calculer la matrice de variance-covariance de Z, de sorte que :

$$\Sigma_z = \begin{bmatrix} f(\mathbf{y}_1, \mathbf{y}_1) & f(\mathbf{y}_1, \mathbf{y}_2) & \cdots & f(\mathbf{y}_1, \mathbf{y}_n) \\ f(\mathbf{y}_2, \mathbf{y}_1) & f(\mathbf{y}_2, \mathbf{y}_2) & \cdots & f(\mathbf{y}_2, \mathbf{y}_n) \\ \vdots & \vdots & \ddots & \vdots \\ f(\mathbf{y}_n, \mathbf{y}_1) & f(\mathbf{y}_n, \mathbf{y}_2) & \cdots & f(\mathbf{y}_n, \mathbf{y}_n) \end{bmatrix}.$$

Il ne reste ensuite plus qu'à calculer les valeurs propres et vecteurs propres ψ_ℓ associées à cette matrice sans avoir à expliciter l'application Φ. On pourra ensuite réduire la dimension des champs par projection linéaire sur l'espace latent de la même façon qu'avec l'ACP standard (**Eq.** 1).

Il existe différentes forme de noyaux reproduisants, en voici quelques unes :

— **Gaussien** : $e^{-\gamma \|\mathbf{y}_i - \mathbf{y}_j\|^2},\ avec\ \gamma = \frac{1}{2\sigma^2}$
— **Polynomial** : $(\gamma \langle \mathbf{y}_i, \mathbf{y}_j \rangle + r)^d$, avec r une constante et d le degré du polynome
— **Sigmoid** : $\tanh(\gamma \langle \mathbf{y}_i, \mathbf{y}_j \rangle + c)$, avec γ le coefficient de la pente et c une constante
— **Cosine similarity** : $\frac{\langle \mathbf{y}_i, \mathbf{y}_j \rangle}{\|\mathbf{y}_i\| \|\mathbf{y}_j\|}$

1. https://fr.wikipedia.org/wiki/ThÃľorÃĺme_de_Mercer
2. https://members.cbio.mines-paristech.fr/~jvert/teaching/2006master/slides/2_noyaux/noyaux.pdf

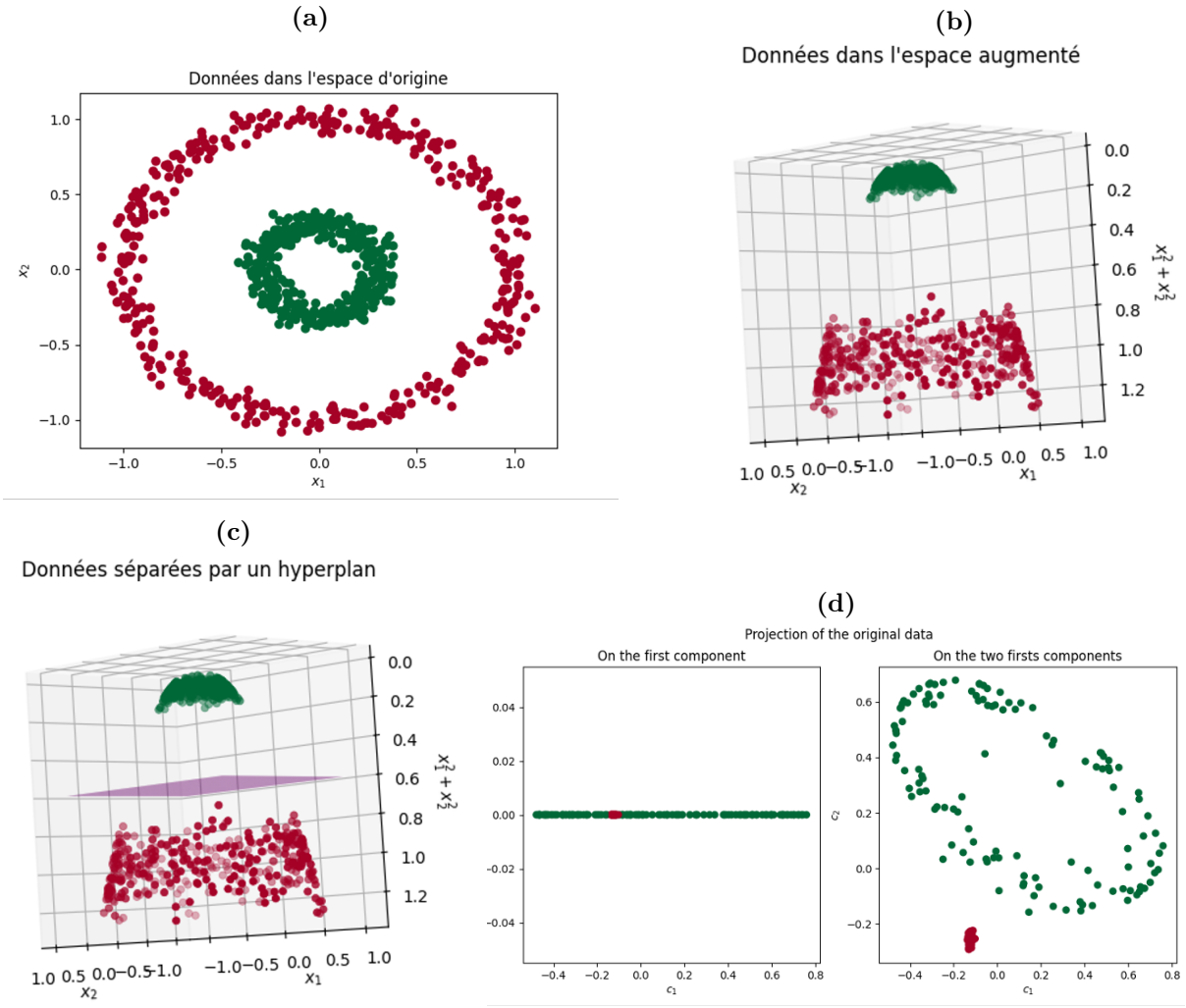


Figure 12 – *Principe de la KPCA réalisé par DELAR avec la fonction KernelPCA de la librairie scikit-learn de python. (a) Représentation des données dans leur espace d'origine dans $\mathbb{R}^2$. (b) Représentation des données dans un espace de dimension supérieur $\mathcal{H}^3$. (c) Représentation d'un hyperplan permettant de séparer les données dans leur nouvel espace de représentation. (d) Résultats de l'application de l'ACP par noyau Gaussien (γ=10, α=0.1) sur les données tests qui permets à présent de discriminer les 2 groupes. A gauche à partir de la 1er composante et à droite à partir des 2 premières composantes.*

En plus de mieux aborder les non-linéarités dans la base d'apprentissage, la méthode KPCA permet de conserver les propriétés principales de l'ACP comme l'orthogonalité des axes deux à deux (qui justifie le fait d'utiliser un processus gaussien indépendant par axe), et la hiérachisation des axes par quantité de variance expliquée. Cependant, le nombre d'hyperparamètres à calibrer est plus important qu'avec l'ACP standard et dépend du noyau choisi.

Par construction, les premiers axes retourné par la POD et l'ACP à noyau ne sont pas forcément ceux qui explique le mieux les champs d'intérêt $\mathbf{y}$. La méthode des moindres carrés partiels tente de répondre à ce problème en adoptant une approche similaire à celle de la POD.

2.1.2 La méthode des moindres carrés partiels (PLS)

L'algorithme des moindres carrés partiels (PLS en anglais) est particulièrement adapté aux ensemble de données de dimension pour lesquels on dispose de peu de réalisations [Bouhlel et al., 2016] et/ou lorsque le nombre de variable est nettement supérieur au nombre d'observation. L'objectif de cette méthodes est proche de celle de l'ACP, puisqu'il s'agit de maximiser la variance expliquée par la base réduite. Mais contrairement à l'ACP, la PLS maximise la covariance entre les données entrées $\boldsymbol{\theta}$, ici les paramètres de conditions météorologiques, et les sorties $\mathbf{y}$, ici le champ de concentration.

Avec cette méthode les axes sont construits un à un de sorte à ce que l'axe suivant soit capable d'expliquer au mieux la part d'information que l'axe précédent n'a pas pu récupérer. Le fonctionnement de la PLS consiste en la résolution du problème d'optimisation suivant [Cadoret, 2024] :

Chercher une suite de D composantes deux à deux orthogonales $\left[C_1, C_2, ..., C_D\right] \in \mathbb{R}^{N_{train} \times D}$ avec $C_d = \boldsymbol{\theta} a_d$ tel que :

$$\max_{\|a_d\|=1} \quad \mathrm{Cov}(\mathbf{y}, \boldsymbol{\theta} a_d)$$

Les champs d'intérêt $\mathbf{y}$ sont, dans cette méthode, projetés sur la base formée par les composantes $\mathbf{C} = (C_1, ..., C_D)$ en apprenant les coefficients $\boldsymbol{\alpha}$ de la relation linéaire entre les composantes C_d et le champs d'intérêt $\mathbf{y}$.

Cette relation linéaire est donnée par :

$$\mathcal{T}(\mathbf{y}) = \sum_{d=1}^{D} \boldsymbol{\alpha}_d C_d + \textbf{résidu}. \tag{5}$$

Il est à noter que Bouhlel et al. [2016] a montré qu'en très grande dimension avec peu d'observation, coupler la PLS et la régression sur processus gaussien permet de construire un modèle réduit en un temps significativement court tout en gardant une bonne précision, ce qui motive que la PLS pourrait être utilisée pour remplacer l'ACP dans le modèle réduit POD–GPR.

2.2 Méthodes d'apprentissage profond

L'apprentissage profond est une méthode d'apprentissage automatique qui s'inspire du fonctionnement des neurones dans un cerveau. Pour reproduire ce fonctionnement, on empile successivement plusieurs couches de neurones artificiels. Cette méthode permet à une machine d'apprendre à réaliser des tâches tel que la vision par ordinateur avec par exemple la détection de cancer à partir d'IRM, le traitement du langage avec par exemple les différents agents conversationnels présent dans notre quotidien ou encore le traitement du signal utilisé en finance pour prévoir la volatilité des indices boursiers. Pour mieux comprendre comment fonction l'apprentissage profond, il est essentiel de formaliser le concept de réseau de neurones artificiels ainsi que les étapes de son apprentissage.

Une couche de réseau de neurones est constituée d'un ensemble de P neurones. Les neurones de la couche reçoivent une entrée et renvoient chacun une transformation différentes de celle-ci. Cette transformation est réalisé à partir de poids et de biais propre à chaque neurones. Le résultat obtenu sera évalué par une fonction d'activation non-linéaire.

Définition 1 (Réseau de neurones artificiels)**.** *Une couche de réseau de neurones est une application définit par :*

$$\begin{aligned} L : &\mathbb{R}^n \to \mathbb{R}^m \\ &\mathbf{x} \mapsto \sigma(\mathbf{x}\mathbf{W} + \mathbf{b}) \end{aligned}$$

Avec $\mathbf{x}$ *l'entrée,* $\mathbf{W}$ *la matrice de poids,* $\mathbf{b}$ *le vecteur de biais et* $\sigma(.)$ *une fonction d'activation non-linéaire.*

Un réseau de neurones est la succession de plusieurs couches de neurones connectés successivement.

Remarque. *Il existe différente fonction d'activation en voici quelques unes qui reviennent de façon récurrente dans la littérature :*

- ***Softmax*** *:* $\frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$ *, permet de transformer les sorties du réseaux en vecteur de probabilités.*
- ***Rectified Linear Unit (ReLU)*** *:* $\max(0, z)$ *, permet de transformer les valeurs négative en 0.*
- ***Tangente hyperbolique (tanh)*** *:* $\frac{e^z - e^{-z}}{e^z + e^{-z}}$ *, permet de transformer les données dans l'intervalle* $(-1, 1)$

Pour optimiser la matrice $\mathbf{W}$ et le vecteur de biais $\mathbf{b}$, l'objectif sera de minimiser une fonction de perte [1] avec la méthode de descente de gradient. La rétro-propagation du gradient [2] est la technique la plus courante en apprentissage profond pour l'optimisation de ces paramètres ($\mathbf{W} et \mathbf{b}$). Elle permet d'actualiser l'ensemble des poids du réseaux de façon simultanée en commençant par calculer le gradient de la couche la plus profonde à celui de la première couche du réseau de neurones - comment optimiser poids - interet (citer papier)

2.2.1 Les auto-encodeurs convolutifs (CAE)

Le principe des auto-encodeurs convolutifs (CAE en anglais) est d'entrainer un réseau de neurones à compresser puis décompresser le plus fidèlement possible une donnée passée en entrée à partir d'une architecture basée sur des couches de convolution (**Fig.** 20). La convolution est une étape clé dans un auto-encodeurs car elle permet de réduire drastiquement le nombre de paramètres à optimiser dans le réseau.

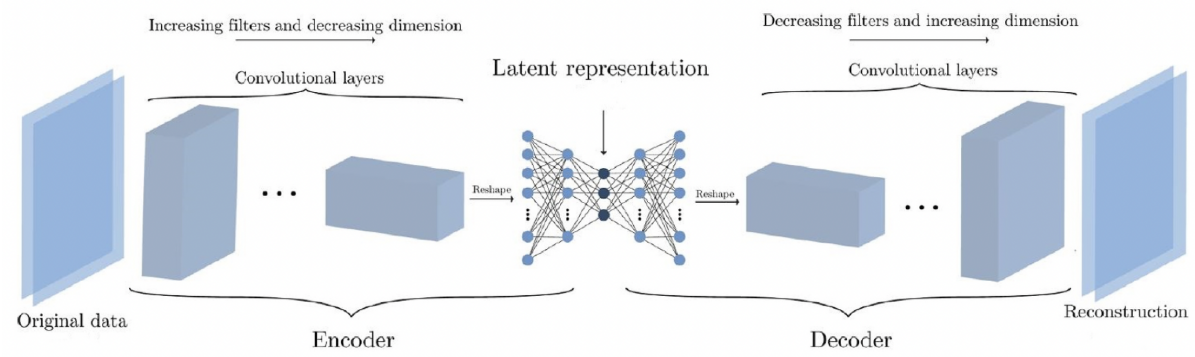


Figure 13 – *Schéma d'une architecture classique d'auto-encodeurs convolutifs par [Masoumi-Verki et al., 2022]*

Après un produit de convolution, on effectue généralement une étape de sous-échantillonnage [3] qui consiste a agréger les valeurs plusieurs région du domaine en effectuant par exemple une moyenne (AveragePooling) ou en retournant la valeur la plus élevé (MaxPooling), avant de passer le résultat dans une fonction d'activation.

1. `https://lrouviere.github.io/machine_learning/cours.pdf`
2. `http://chercheurs.lille.inria.fr/pgermain/neurones2019/04-retropropagation.pdf`
3. `https://cedric.cnam.fr/vertigo/cours/ml2/coursDeep3.html#sous-echantillonnage`

Cet enchaînement est répété plusieurs fois avec une modification possible des hyperparamètres (dimension de la zone du domaine à agréger à l'étape de sous-échantillonnage, méthode utilisée pour les agréger, choix de la fonction d'activation) entre chaque couche de convolution.

Pour finir, les données sont passées à travers des couches de réseau entièrement connectées, contenant de moins en moins de paramètres en fonction de la dimension souhaitée pour l'espace latent. Ce principe représente la partie encodeur, et le résultat obtenu est notre espace latent, que l'on va décompresser avec un décodeur qui effectue le même type d'opération (convolutions transposées [1] et sur-échantillonnage [2]).

L'utilisation de réseaux de neurones pour la construction de modèle réduits de dynamique des fluides permet de casser la barrière que les méthode linéaire n'arrive pas à franchir [Gruber et al., 2022a, Murata et al., 2020, Nony, 2023]. De nombreux travaux ont également montrés que les reseaux de neurones convolutifs (CAE en anglais) ont des performances largement supérieur à la POD pour agréger des données dans un espace latent [Murata et al., 2020, Masoumi-Verki et al., 2022]. De plus, il est possible, comme pour la POD, d'avoir une décomposition hiérarchique des axes de l'espace latent en fonction de l'information qu'elles portent [Fukami et al., 2020]. On peut alors de quantifier l'information apporté par chaque axes de l'espace latent et donc éliminer les axes qui ajouteraient du bruit dans notre modèle.

Limites Le produit de convolution ne peut être appliqué que sur des données structurées, il est nécessaire d'interpoler nos champs exprimés sur un maillage non structuré sur un nouveau maillage structuré. Cette étape nécessite d'augmenter la taille des champs ce qui ajoute inévitablement une erreur d'interpolation. De plus, plus le réseau est profond plus il nécessite un nombre conséquent de données pour l'optimisation de ces paramètres. Cependant, le temps de calcul élevés du modèle LES pour générer des observations limite le choix de la génération de nouvelle données.

2.2.2 Quelques auto-encodeurs pour les graphes

Pour palier la contrainte d'application aux données structurées des CAEs, j'ai recherché dans la littérature des méthodes de réduction de dimension avec architecture de type auto-encodeur permettant de traiter des graphes. En effet, les graphes sont des objets adaptés à la représentation de maillages non structurés comme celui du modèle MUST (**Fig.** 3). Je présente dans la suite deux méthodes que j'ai identifiées comme étant particulièrement pertinentes même si je n'ai pas eu le temps d'aller jusqu'à leur implémentation pour notre cas d'étude.

Auto-encodeur convolutif basé sur deux courbes de Hilbert (SFC-CAE)

Le principe de l'auto-encodeur convolutif pour graphes introduit par [Heaney et al., 2024] (SFC-CAE pour *convolutional autoencoder with two space-filling curves*) est d'utiliser deux courbes de Hilbert [Nguyen, 2013] pour transposer de façon ordonnée les données du graphe, ici les champs discrétisés sur le maillage non structuré, dans un vecteur 1D structuré représentatif des distances dans le graphe (**Fig.** 14). Ensuite, on applique l'algorithme des plus proches voisins au vecteur 1D, ce qui permet d'appliquer un auto encodeur classique avec des couches de convolution 1D.

Remarque. *L'utilisation de deux courbe de Hilbert est motivé par le fait que certains noeuds du domaine qui sont extrêmement proche ont un risque d'être éloigné lors de l'ordonnement [Heaney et al., 2024].*

1. `https://cedric.cnam.fr/vertigo/cours/ml2/coursDeep3.html#convolution-transposee`
2. `https://cedric.cnam.fr/vertigo/cours/ml2/coursDeep3.html#sur-echantillonnage`

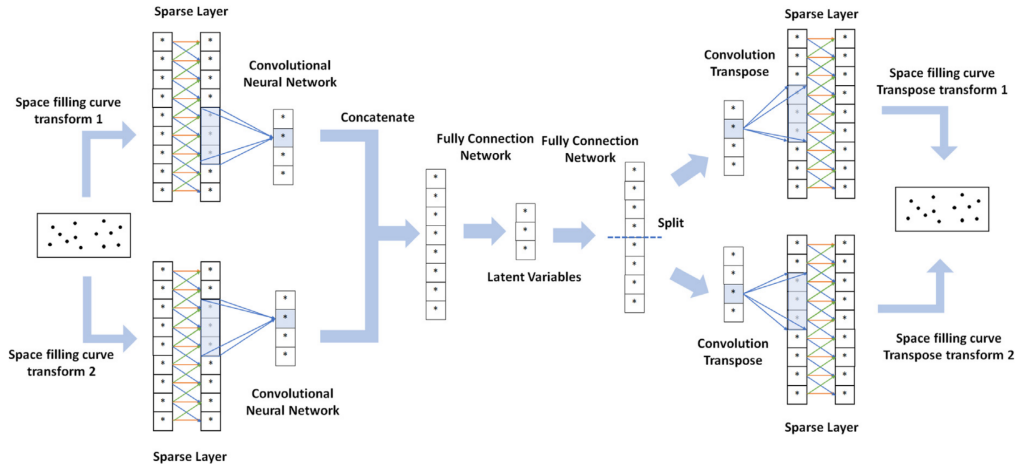
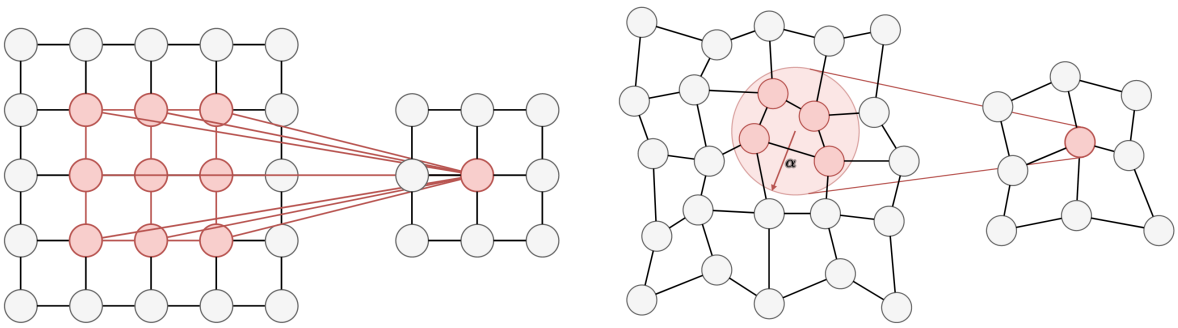


Figure 14 – *Architecture de la SFC-CAE par [Heaney et al., 2024].*

Cette méthode semble particulièrement judicieuse si l'on souhaite tester un jour l'apprentissage par transfert. En effet on pourrait d'abord entraîner le réseau sur une ou plusieurs coupes horizontales avant de finir l'entraînement sur le champ complet, afin de bénéficier des similarités dans la structure du panache. En effet, l'architecture proposée (**Fig.** 14) ne dépend pas de la taille des données d'entrées. Il faudrait donc juste construire des courbes de Hilbert plus grandes pour le champ complet et ensuite appliquer les mêmes couches de convolution au champ vectorisé en 1D.

Auto-encodeurs convolutifs basées sur la quadrature

Durant ses travaux, [Doherty et al., 2024] a montré qu'il est possible d'effectuer une convolution continue directement sur un maillage non structuré. Cette convolution est basé sur des méthodes de quadrature, des techniques pour effectuer une intégration numérique. Elle permet d'approximer avec une intégrale la convolution discrète sur des points du maillage(**Fig.** 15).



(a) Discrete convolution with 3x3 kernel on uniform grid.

(b) QuadConv with α compact support on non-uniform mesh.

Figure 15 – *Représentation du principe de convolution (a) et du principe de la convolution par quadradure (b) par [Doherty et al., 2024].*

La méthode QCAE (pour *Quadrature-based convolutions autoencoder*) proposée possède une architecture classique d'auto-encodeurs(**Fig.** 16).

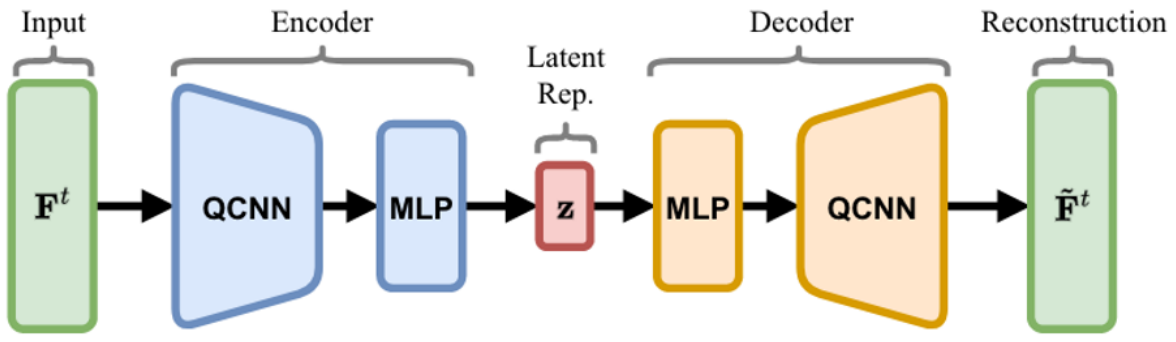


Figure 6: Autoencoder structure.

Figure 16 – *Schéma de l'architecture du QCAE par [Doherty et al., 2024].*

Cette méthode est particulièrement intéressante car elle donne la possibilité d'appliquer le QCAE à des maillages structurés. Il serait donc possible de comparer les résultats du QCAE et du CAE (Section [2.2.1]) dans une future implémentation. Lors du processus de validation de sa méthode, [Doherty et al., 2024] a montré que la QCAE donne de meilleur performance que la POD, ce qui rend cette méthode d'autant plus pertinente.

3 Bilan des résultats obtenus

Cette section présente les résultats obtenus avec les différentes méthodes de réduction de dimension testées. Pour que leur inter-comparaison soit équitable, nous avons fixé la dimension de l'espace latent à dix composantes, soit le nombre de composante retenu dans le modèle réduit de référence POD–GPR.

3.1 Démarche de comparaison

Dans cette section, je présente comment j'ai évalué la précision de chacune des différentes méthodes de réduction de dimension.

3.1.1 Simplifications

Par soucis d'efficacité, la base de données que j'ai utilisé contenait uniquement les coupes horizontal à 1m60 des champs 3-D retournés par le modèle LES. On travaille maintenant sur la coupes 2-D de chaque champs. Grâce à cela j'ai pu réduire considérablement le temps de calibration de chaque méthodes. De plus, un autre moyen d'optimiser l'apprentissage est de remettre l'ensemble des données à la même échelle. Pour ça, j'ai utilisé la transformation **Eq.** 2) qui donne les meilleurs résultats.

Aussi, afin de comparer équitablement l'ensemble des méthodes, j'ai fixé la dimension de l'espace latent pour chaque méthode à 10. Ainsi, nous pouvons comparer chacune d'entre elles à la méthode de référence, la POD.

3.1.2 Erreur de réduction-reconstruction

L'objectif premier de mes travaux durant le stage est de corriger les erreurs liées à l'étape de réduction de dimension du modèle POD-GPR en trouvant une alternative à l'ACP (Section [1.5]).

Nous avons vu que les erreurs commise par le modèle sont principalement dû à la reconstruction. En effet, après avoir encoder les données dans un espace latent de plus petite dimension, la perte d'information que l'on a dépend dans notre cas des 10 premières composantes. La reconstruction de l'espace se faisant à partir de cet espace latent, il est essentiel que les 10 composantes aient agrégés suffisamment d'information pour commettre le moins d'erreur possible. L'erreur de reconstruction affecte l'entièreté des prédiction du modèle réduit. Dans ce contexte, il est primordiale de comparer les erreurs de reconstruction liée à chacune des méthodes avec celle initialement faite par la POD. Pour la suite de cette section, seules les erreurs de reconstruction des différentes méthodes seront étudiées.

3.1.3 Les métriques de validation

Dans cette section, nous définissons les différentes métriques utilisées pour valider les prévisions des différents modèles de réduction de dimension. Il s'agit des métriques standard de la qualité de l'air introduites par Chang and Hanna [2004] qui comparent les valeurs de concentration prédites par le modèle réduit c_{ROM} et les valeurs de référence du modèle LES c_{LES} :

— La variance géométrique (VG) est à valeur dans $\mathbb{R}^+$, et la valeur optimale recherché est 1. Elle mesure l'écart relatif entre les observations et les prévisions. Un fort écart entre ces deux valeurs se traduit par une valeur élevée de VG, résultant soit d'un biais, soit d'une forte variance, ou d'une combinaison des deux.

$$\mathrm{VG} = \exp\left(\left\langle (\ln \widetilde{\mathbf{c}}_{\mathbf{LES}} - \ln \widetilde{\mathbf{c}}_{\mathbf{ROM}})^2 \right\rangle\right) \tag{6}$$

— L'erreur quadratique moyenne normalisée (NMSE) est à valeur dans $\mathbb{R}$, et la valeur optimale recherché est 0, elle donne une mesure de la précision global du modèle. Dans le cas où la NMSE prend de forte valeur correspond à la présence d'un biais élevé ou à une variance élevé des erreurs fort.

$$\text{NMSE} = \frac{\langle(\bar{\mathbf{c}}_{\mathbf{LES}} - \bar{\mathbf{c}}_{\mathbf{ROM}})^2\rangle}{\langle\bar{\mathbf{c}}_{\mathbf{LES}}\rangle\langle\bar{\mathbf{c}}_{\mathbf{ROM}}\rangle} \tag{7}$$

— La proportion des concentrations prédites comprises à un facteur deux des observations (FAC2). Cette métrique est naturellement comprise en 0 et 1, plus la FAC2 tend vers 1 meilleure est la modélisation. Chacune des valeurs à la même importance, cette méthode est robuste pour les valeurs extrême et renvoie une information précise sur les performances du modèle.

$$\text{FAC2} = \frac{1}{N_{obs}} \sum_{k=1}^{N_{obs}} \varepsilon_k \text{ avec } \varepsilon_k = \mathbb{1}_{(0.5 \leq \bar{c}_{ROM}^{(k)}/\bar{c}_{LES}^{(k)} \leq 2) \vee (\bar{c}_{ROM}^{(k)} \leq c_t \text{ et } \bar{c}_{LES}^{(k)} \leq c_t)} \tag{8}$$

L'utilisation de plusieurs métriques permet de mieux aborder la qualité des prévisions notamment compte tenu du grand éventail d'ordre de grandeurs de concentration en jeu [Chang and Hanna, 2004]. Rappelons que les performances des modèles réduits sont limité par celles du modèle LES. Autrement dit, les valeurs optimales des modèles réduits seront bornées par la variabilité interne du modèle LES. La valeur optimale pour la pour la VG est de 1.39, de 1.80 pour la NMSE et de 0.95 pour la FAC2 [Lumet, 2024]

3.2 Calibration des méthodes de réduction

Nous verrons dans cette sections les résultats obtenues pour les différentes méthodes que j'ai implémentée et la démarche que j'ai utilisée pour calibrer les méthodes paramétrique (ACP à noyau et auto-encodeurs convolutifs).

Les méthodes PLS et ACP à noyau ont été mis en œuvre en python à partir de la librairie Scikit-Learn. La méthode par auto-encodeurs convolutif a elle été implémenté avec la librairie PyTorch de python. Les codes ont quant à eux étés exécutés sur le supercalculateur KRAKEN du CERFACS.

3.2.1 Approche PLS

Pour calibrer la méthode PLS, nous avons uniquement besoin de définir le nombre de dimensions souhaité pour l'espace latent. Dans le but de comparer les nouvelles approches de réduction de dimension avec la POD, j'ai initialement fixé ce nombre à 10. Il n'y avait donc aucun hyperparamètre à calibrer pour cette méthode. De même pour les méthodes suivante, le nombre de composantes pour l'espace latent est fixé à 10.

3.2.2 Approche ACP à noyau (KPCA)

L'ACP à noyau est une méthode paramétrique en fonction du noyau f^* utilisées. Pour la méthode d'ACP à noyau, la fonction KernelPCA [1] de la librairie Scikit-Learn propose 5 fonctions noyaux pré-implémenté, le noyau 'linear' qui permet d'effectuer une ACP classique, le noyau 'poly' pour Polynomiale, le noyau 'rbf' pour Gaussien, 'sigmoid' pour Sigmoid et 'cosine' pour la similarité cosine. On pourra retrouver leur forme explicite en Section [2.1.1]. Il est également possible d'implémenter son propre noyau avec l'option 'precomputed', cependant cette piste n'a

1. `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.KernelPCA.html`

pas été exploré dans le cadre de mon stage. Chaque noyau f^* possède des hyperparamètres Λ^* différents à calibrer. Pour calibrer efficacement ces hyperparamètres, j'ai utilisé une validation croisée 3-blocs [1].

Cette méthode consiste à partitionner de façon aléatoire la base d'entraînement en 3 blocs $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ de 50 réalisations LES, puis a entraîner le modèle sur deux d'entre eux. Une fois le modèle calibré, il sera évalué avec la métrique FAC2 sur l'échantillon qui n'a pas servis lors de l'entraînement **Fig.**17. La validation s'arrête une fois que chacun des blocs a été utilisé une fois comme échantillon. Le résultat retourné est la moyenne des erreurs faites lors de la boucle de validation.

Au final, l'ensemble des combinaisons Λ^* pour chacun des noyaux f^* (rbf, polynomial, sigmoid, cosine) ont été évaluées et classées selon le critères de FAC2. Les meilleurs combinaisons d'hyperparamètres retenues selon le critères de FAC2 sont listé dans le tableau 1 :

Paramètre KernelPCA	**RBF1**	**RBF2**	**Poly**	**Sigmoid**	**Cosine**
kernel	rbf	rbf	poly	sigmoid	cosine
alpha	$1e-6$	$1e-7$	5	0.005	0.0413
gamma	0.01	0.02	0.8	0.09045	None
degree	3	3	1	3	3
coef0	1	1	10	0.0002	1
FAC2	**0.951**	**0.96**	**0.891**	**0.841**	**0.886**

Table 1 – *Score FAC2 obtenus par la KPCA avec les hyperparamètres Λ^* optimisés par validation croisée 3-blocs*

J'ai par la suite choisi de garder deux de ces combinaisons d'hyperparamètres nommées RBF1 et RBF2. Elles prennent chacune des valeurs de FAC2 supérieur à 0.95, avec 0.951 pour RBF1 et 0.96 pour RBF2. Bien que RBF2 montre de meilleurs résultats par rapport à la combinaisons RBF1 en terme de FAC2, cette dominance s'inverse pour le critère de NMSE avec une NMSE de 5.75 pour RBF1 contre de 13.58 pour RBF2.

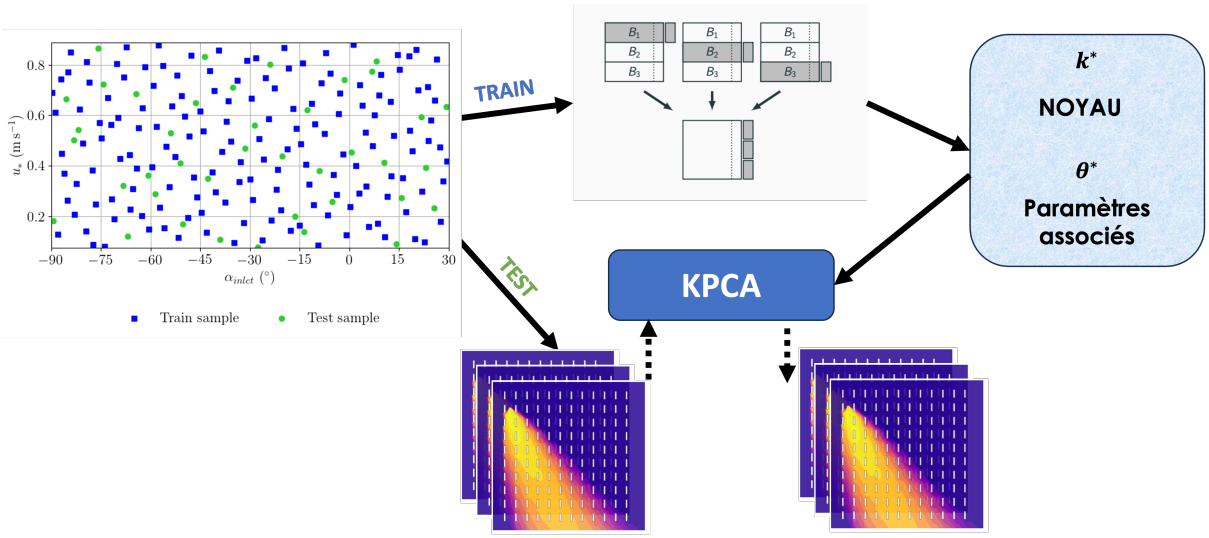


Figure 17 – *Représentation schématique du protocole de validation utilisé pour la calibration des hyperparamètres de l'ACP à noyau*

1. https://lrouviere.github.io/machine_learning/cours.pdf

3.2.3 Approche auto-encodeurs convolutifs

Pour calibrer mon CAE, je me suis basé sur deux architectures, celle de [Nony, 2023] qui propose une application d'auto-encodeurs convolutifs à des données généré avec un modèle LES également pour la dispersion de polluant en milieu urbain ainsi que celle de [Xiang et al., 2021] qui propose une architecture d'auto-encodeurs pour la réduction de dimension pour de la simulation de vent en milieu urbain. L'apprentissage par réseau de neurones ayant ses propres spécificités, j'ai dû effectuer un nouveau partitionnement de la base de données en utilisant cette fois, 150 données pour l'apprentissage, 40 pour la phase de test et 10 pour la validation du CAE **Fig.**18.

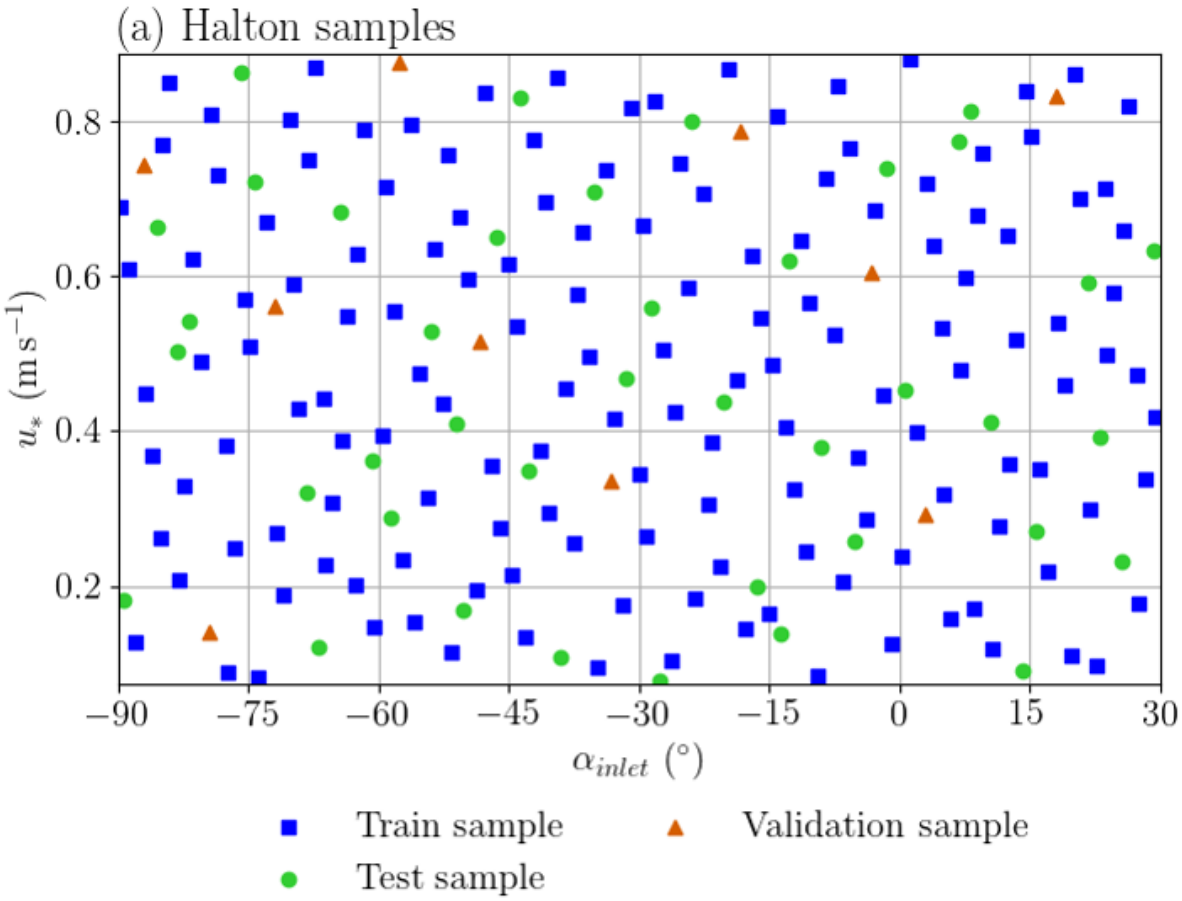


Figure 18 – *Répartition des conditions limites dans l'espace des paramètres associées à chaque champs pour les différentes étape de calibration du CAE.*

Durant mon stage, j'ai pu implémenter chacune de ses architectures, cependant elles avaient toutes les deux des difficultés à converger (**Fig.**19).

Les différentes architectures de réseau ne permettaient donc pas d'effectuer des prédictions. Pour optimiser la phase d'entraînement, je me suis donc tourné vers différentes techniques. La première consiste à utiliser un coefficients réduisant la vitesse d'apprentissage du réseau à chaque boucle d'apprentissage. Ce qui permet de stabiliser la convergence des poids. J'ai également essayé différentes configuration en modifiant les fonctions d'activation, la dimension des noyau de convolution ainsi que celle des couches de neurones, la méthode de descente de gradient et la méthode de pré-traitement des données.

Finalement, j'ai retenu l'architecture de [Xiang et al., 2021], en ayant au préalable centrée

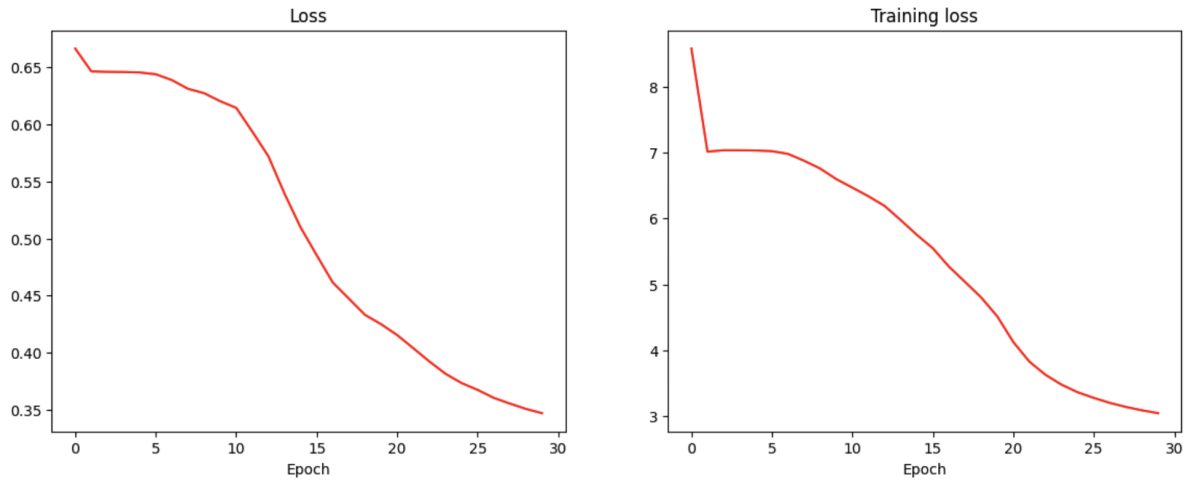


Figure 19 – *Exemple de fonction de perte (Loss) (à gauche) et de fonction de perte d'entraînement qui évalue l'ajustement du modèle à chaque boucle d'apprentissage (à droite) qui ne convergent lors de l'entraînement d'un CAE.*

et réduit les données d'entrée et utilisé l'optimiseur Adam [1] avec un taux d'apprentissage de $7e-04$ et un coefficient décroissant de 0.93. Tout cela à considérablement amélioré les résultats, cependant l'architecture de CAE (**Fig.** 20) que j'ai implémenté est toujours améliorable.

3.3 Inter-comparaisons des modèles

Erreurs de reconstruction La comparaison des différentes méthodes de réductions de dimensions s'est fait selon les critères de métriques définit précédemment en Section [3.1.3]. Pour l'ensemble des méthodes de réduction de dimension, j'ai évalué les erreurs commises lors de la reconstruction d'une coupe d'intérêt. Ces résultats sont listé dans le tableau 2 :

Métriques	POD	RBF1	RBF2	PLS	CAE
VG	**1.172**	**1.124**	**1.125**	**1.190**	**1.493**
NMSE	**38.328**	**5.752**	**13.579**	**65.028**	**156.518**
FAC2	**0.925**	**0.951**	**0.960**	**0.917**	**0.879**

Table 2 – *Score VG, NMSE et FAC2 obtenus pour chaque méthodes de réductions de dimension retenue.*

On constate que la PLS, qui est également une technique de réduction de dimension linéaire, obtient des résultats similaires à ceux de la POD. La différence entre ces deux méthodes se fait essentiellement sur le critère NMSE. Cependant, ce critère peut être influencé par de nombreux facteurs, notamment par la reconstruction au niveau de la source d'émission, qui est une zone à forte concentration que les modèles ont du mal à reproduire. Toutefois, cela montre que les informations fournies par les conditions limites associées à chaque coupe n'apportent pas de valeur ajoutée significative pour réduire plus efficacement leur dimension.

L'utilisation de noyaux permettant la transformation de l'ACP classique en une ACP non-linéaire améliore très significativement les résultats de la POD sur l'ensemble des métriques. On peut donc maintenir l'hypothèse que la non-linéarité de la réponse du champ de concentration

1. `https://pytorch.org/docs/stable/generated/torch.optim.Adam.html`

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #                  Kernel Shape
==========================================================================================
CAE                                      [1, 1, 512, 512]          --                       --
├─Encodeur: 1-1                          [1, 128, 14, 14]          --                       --
│    └─Conv2d: 2-1                       [1, 64, 492, 492]         7,808                    [11, 11]
│    └─MaxPool2d: 2-2                    [1, 64, 246, 246]         --                       2
│    └─ReLU: 2-3                         [1, 64, 246, 246]         --                       --
│    └─Conv2d: 2-4                       [1, 64, 234, 234]         200,768                  [7, 7]
│    └─MaxPool2d: 2-5                    [1, 64, 78, 78]           --                       3
│    └─ReLU: 2-6                         [1, 64, 78, 78]           --                       --
│    └─Conv2d: 2-7                       [1, 128, 74, 74]          73,856                   [3, 3]
│    └─MaxPool2d: 2-8                    [1, 128, 37, 37]          --                       2
│    └─ReLU: 2-9                         [1, 128, 37, 37]          --                       --
│    └─Conv2d: 2-10                      [1, 128, 33, 33]          147,584                  [3, 3]
│    └─MaxPool2d: 2-11                   [1, 128, 16, 16]          --                       2
│    └─ReLU: 2-12                        [1, 128, 16, 16]          --                       --
│    └─Conv2d: 2-13                      [1, 128, 14, 14]          147,584                  [3, 3]
│    └─ReLU: 2-14                        [1, 128, 14, 14]          --                       --
├─Flatten: 1-2                           [1, 25088]                --                       --
├─Linear: 1-3                            [1, 10000]                250,890,000              --
├─Linear: 1-4                            [1, 10]                   100,010                  --
├─Linear: 1-5                            [1, 10000]                110,000                  --
├─Linear: 1-6                            [1, 25088]                250,905,088              --
├─Decodeur: 1-7                          [1, 1, 512, 512]          --                       --
│    └─Unflatten: 2-15                   [1, 32, 28, 28]           --                       --
│    └─ConvTranspose2d: 2-16             [1, 64, 86, 86]           100,416                  [7, 7]
│    └─ReLU: 2-17                        [1, 64, 86, 86]           --                       --
│    └─ConvTranspose2d: 2-18             [1, 16, 256, 256]         9,232                    [3, 3]
│    └─ReLU: 2-19                        [1, 16, 256, 256]         --                       --
│    └─ConvTranspose2d: 2-20             [1, 1, 772, 772]          785                      [7, 7]
│    └─Upsample: 2-21                    [1, 1, 512, 512]          --                       --
==========================================================================================
Total params: 502,693,131
Trainable params: 502,693,131
Non-trainable params: 0
==========================================================================================
Input size (MB): 10.00
Forward/backward pass size (MB): 2504.35
Params size (MB): 1917.62
Estimated Total Size (MB): 4431.97
==========================================================================================
```

Figure 20 – *Résumé de l'architecture auto-encodeurs du modèle CAE*

moyen aux variations des paramètres d'entrée est à l'origine des erreurs commises par la POD. N'ayant pas pu terminer l'optimisation du CAE, ce résultat motive la pertinence de l'approfondissement de cette méthode, qui, pour le moment, donne des résultats moins bons que ceux de la POD, mais qui est prometteuse.

En effet, le modèle d'auto-encodeurs convolutifs que j'ai retenu donne les pires résultats sur l'ensemble des métriques. Or, il a déjà été montré dans la littérature [Nony, 2023, Murata et al., 2020] que pour des cas d'application similaires, cette méthode est capable de donner de bien meilleurs résultats. Cependant, il serait pertinent de quantifier l'impact de l'interpolation du maillage non structuré vers un maillage structuré sur les performances du CAE.

Concernant la taille de la base de données d'apprentissage, nous avons réalisé un test à partir d'une base de données contenant 2000 réalisations du modèle POD–GPR. Cela consiste à utiliser cette nouvelle base de données pour l'entraînement de cette architecture CAE. Les résultats de cette expérience nous ont permis de mettre de côté l'hypothèse concernant les limites de notre base d'apprentissage LES pour entraîner le CAE. Autrement dit, augmenter la taille de la base d'apprentissage n'améliore pas les performances du CAE.

Durée d'apprentissage et de prédiction L'objectif d'un modèle réduit est d'avant tout permettre de pouvoir optimiser un émulateur de modèle plus complexe afin d'effectuer des simulations de sa surface de réponse. Le temps d'entraînement et de prévision associées à chaque méthode est donc un critères à retenir. La méthode la plus rapide pour la phase d'entraînement est la KPCA avec un temps moyen de 0.38 seconde (auxquels il faut ajouter le temps de re-

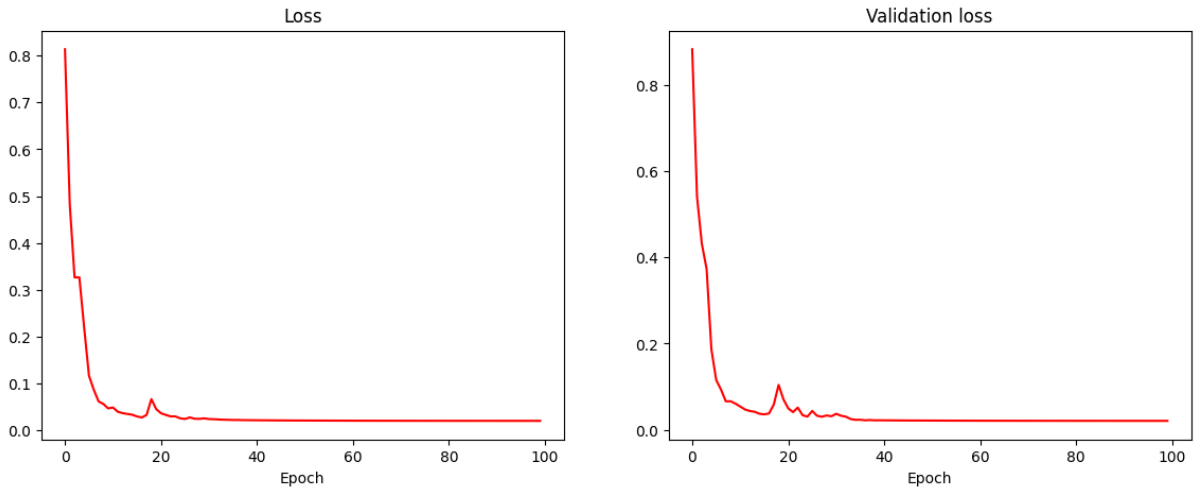


Figure 21 – *Graphique de la fonction de perte (Loss) du réseaux de neurones au cours de l'apprentissage (à gauche) et graphique de l'évaluation du modèle à chaque boucle d'apprentissage sur des données de validation sans mise à jour des poids (à droite)*

cherche des hyperparamètres optimaux ce qui varie en fonction du problème), suivis de la PLS avec un temps moyen de 2 secondes, vient ensuite la POD avec 4.8 secondes en moyenne et enfin le CAE avec 293.72 secondes soit un peu moins de 5 minutes (auxquels il faut ajouter le temps de recherche des hyperparamètres optimaux ce qui varie en fonction du problème).

Méthode retenu J'ai choisis de retenir l'analyse en composantes principales à noyau (KPCA) avec noyau Gaussien (RBF1) (Tableau 1) pour l'étape de réduction de dimension. Les coupes reconstruites par la KPCA avec la configuration RBF1 arrive globalement à bien reproduire la coupes de champs LES de référence (**Fig.** 22).

La **Figure** 22 (c) montre qu'au niveau de la source le modèle sur-estime et sous-estime des concentrations mais que globalement les estimations sont correcte sur l'ensemble du domaine dans cette exemple. Cela pourrait expliquer la NMSE très basse pour cette méthode (Tableau 2), car les erreurs semblent se compenser au niveau de la source.

Un exemple de comparaison (**Fig.** 23) de cette nouvelle méthode avec la méthode POD, référence de mon stage, montre que la KPCA avec configuration RBF1 semble avoir plus de facilité à représenter les concentrations pour les coupes de champs plus complexes orientés vers les structures urbaines.

Je me suis par la suite intéressé a son utilisation pour entraîner les processus Gaussien, ce qui à mené à la création du modèle réduit KPCA–GPR. Les résultats de cette méthode (Tableau 3), ont montré des résultats très nettement supérieur à ceux du modèle POD–GPR. En plus, de bien réduire et reconstruire les coupes, leur représentation de l'espace latent permet également l'amélioration de l'apprentissage des processus Gaussien.

Métriques	POD–GPR	KPCA–GPR
VG	**0.995**	**0.999**
NMSE	**26.141**	**4.244**
FAC2	**0.920**	**0.946**

Table 3 – *Score VG, NMSE et FAC2 obtenus pour les modèles réduits POD–GPR et KPCA–GPR.*

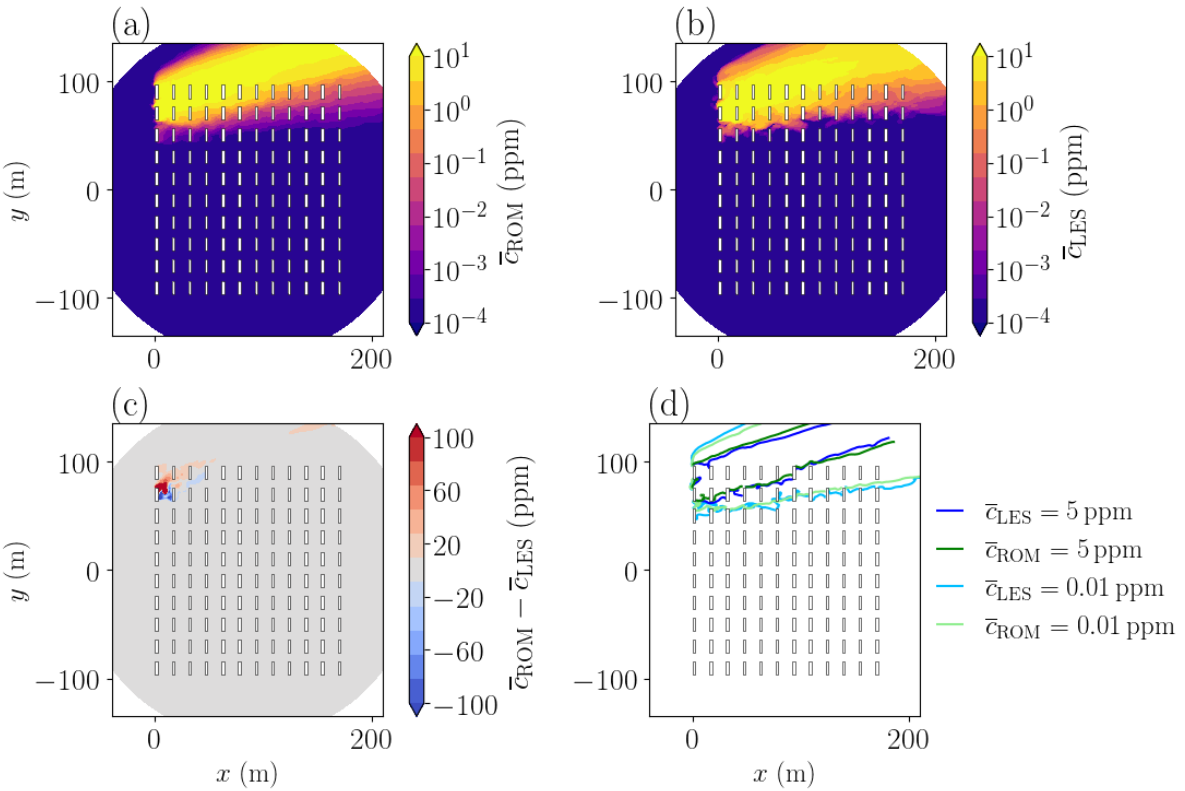


Figure 22 – *Exemple de coupe reconstruite par la KPCA avec la configuration RBF1 (Tableau 1) (a), pour une coupe LES de référence (b). La différence entre les deux modèles (c) et la forme de panache reconstruite par les deux modèles (d)*

3.4 Apprentissage par transfert

L'apprentissage par transfert est un outil du domaine de l'apprentissage automatique qui consiste à utiliser les connaissances acquises lors de l'apprentissage d'une tâche pour améliorer les performances sur une tâche analogue [1]. Nous nous sommes donc naturellement intéresser à cette méthodes pour optimiser l'apprentissage de notre modèle réduit.

Comme mentionné en (Section 3.1.1), nous avons entraîné et évalué le modèle réduit KPCA-GPR sur des coupes 2D des champs de concentration retourné par le modèle LES, pour diminuer le temps nécessaire à la calibration du modèle. Par la suite, nous avons cherchés à savoir si les hyperparamètres retenues en 2D par la recherche par grille sont aussi pertinent en 3D.

Pour cela, une recherche par grille a été effectué sur les champs 3D pour 160 combinaisons d'hyperparamètres contenant celle retenue en 2D. Cette opération à durée environs trois heures et a montrée que la combinaison d'hyperparamètres retenue en 2D se situe dans la zone contenant les combinaisons d'hyperparamètres les plus performant en 3D.

Ce résultat permet d'aborder une nouvelle approche, plus efficiente, pour notre étape d'apprentissage. En effet, le temps de calibration du modèle étant beaucoup plus important en 3D qu'en 2D, ce résultat justifie le fait d'utiliser les hyperparamètres retenus en 2D ou d'effectuer une recherche par grille dans une zone restreinte, proche de celle-ci, pour calibrer le modèle réduit en 3D.

1. `https://www.cnil.fr/fr/definition/apprentissage-par-transfert`

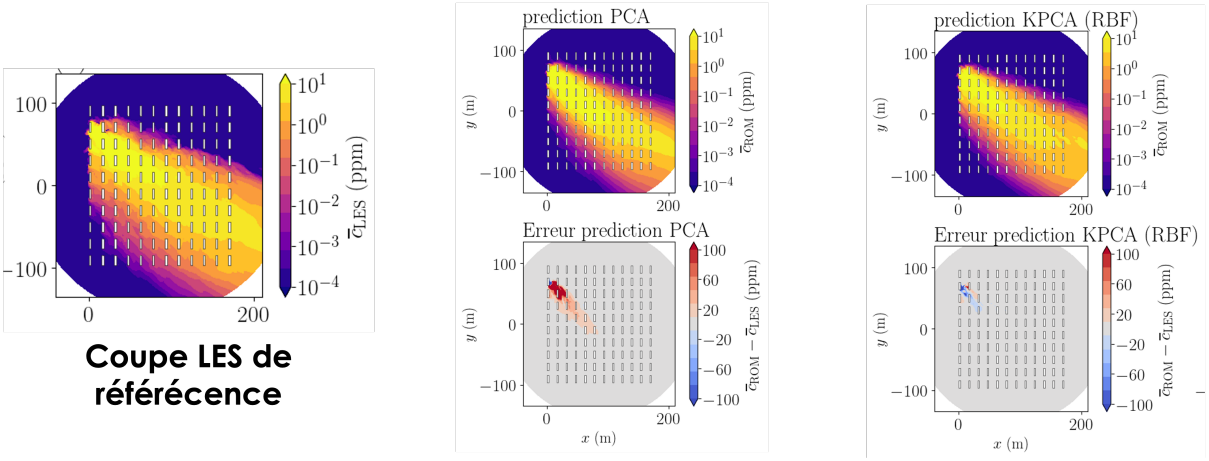


Figure 23 – *Exemple pour une coupe LES de référence (à gauche) la coupe reconstruite par la PCA et sa différence à celui de la coupe de référence LES (au centre) ainsi que la coupe reconstruite par la KPCA et sa différence à celui de la coupe de référence LES.*

Bilan du stage

L'objectif des travaux réalisés lors de mon stage ont pour but de trouver une méthode de réduction de dimension plus performantes que la POD. J'ai pu étudier trois méthodes à savoir, l'analyse en composantes principales à noyau (KPCA), la régression des moindres carrés partiels (PLS) et les auto-encodeurs convolutifs (CAE). Les résultats obtenues ont monté que la KPCA est la méthode la plus efficace en l'état. Ce qui signifie que la prise en compte des non-linéarité de la réponse du champ de concentration moyen aux variations des conditions limites est pertinente pour l'amélioration de l'étape de réduction de dimension. Les résultats de la PLS, autre méthode linéaire, très proches de ceux de la POD appuient cette hypothèse. La possibilité d'effectuer de l'apprentissage par transfert d'une coupe 2-D vers un champ 3-D pour l'optimisation des hyperparamètres peu également permettre de réduire très significativement le temps de calibration de la KPCA pour des données 3-D.

À l'issue de ce stage, plusieurs perspectives intéressantes pour la poursuite des travaux s'offrent à nous. Tout d'abord, l'amélioration du CAE représente une piste de développement non négligeable. À la fin de mon stage, il est apparu que l'architecture du CAE pourrait encore être optimisée, en particulier en adaptant les hyperparamètres du réseau comme les fonctions de perte ou la taille des couches de neurones. De plus, l'implémentation des méthodes SFC-CAE (Section 2.2.2) et QCAE (Section 2.2.2), qui permettent de prendre en compte naturellement la représentation non structurée de nos maillages, constitue une autre piste prometteuse. Ces approches permettraient de gagner en précision car il ne serait plus nécessaire d'interpoler les champs ce qui ajoutait des erreurs à notre CAE.

Un autre aspect important à explorer est l'évaluation de la capacité des processus Gaussien à prédire les coefficients de l'espace latent pour les nouvelles méthodes. Avec les méthodes d'apprentissage profond, la structure de l'espace latent est moins contrôlée comparativement à l'ACP. Il est également important de s'assurer qu'une amélioration dans la réduction de dimension de la méthode POD–GPR n'empêche pas la phase d'apprentissage des processus Gaussien. Ainsi, une analyse approfondie du nouveau modèle réduit complet avec la méthode de réduction de dimension retenu est nécessaire pour garantir la cohérence et la performance globale de la méthode.

Enfin, les résultats obtenus avec la KPCA montrent que les informations apprises sur des coupes horizontales peuvent être transférées efficacement à l'apprentissage des champs complets

en 3D. Il serait donc intéressant d'étendre cette approche en explorant l'apprentissage basé sur un ensemble de coupes à différentes hauteurs, en particulier pour les réseaux de neurones pour lesquels les données peuvent être limitées. Cette méthode pourrait offrir de nouvelles perspectives pour améliorer la qualité et la précision des modèles dans des contextes de données limitées.

Références

Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1) :539–575, 1993. DOI: 10.1146/annurev.fl.25.010193.002543.

C.A. Biltoft. Customer report for Mock Urban Setting Test. DPG Document No. WDTC-FR-01-121, West Desert Test Center, U.S. Army Dugway Proving Ground, Utah, USA, 2001.

B Blocken, Y Tominaga, and Ted Stathopoulos. Cfd simulation of micro-scale pollutant dispersion in the built environment. *Building and Environment*, 64 :225–230, 2013.

Mohamed Amine Bouhlel, Nathalie Bartoli, Abdelkader Otsmane, and Joseph Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53 :935–952, 2016.

Isabelle Cadoret. Econométrie avancée : causalité, lasso, ridge, 2024. Cours donné à l'Université de Rennes pour les étudiants de Master 1 MAS, parcours SDSE, durant le semestre 2.

Joseph C Chang and Steven R Hanna. Air quality model performance evaluation. *Meteorology and Atmospheric Physics*, 87(1) :167–196, 2004.

Pierre-Andre Cornillon. Analyse de données, 2024. Cours donné à l'Université Rennes 2 pour les étudiants de Master 1 MAS, durant le semestre 2.

Hunor Csala, Scott T. M. Dawson, and Amirhossein Arzani. Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. *Physics of Fluids*, 34(11) :117119, 2022. ISSN 1070-6631. DOI: 10.1063/5.0127284. URL `https://doi.org/10.1063/5.0127284`.

Kevin Doherty, Cooper Simpson, Stephen Becker, and Alireza Doostan. Quadconv : Quadrature-based convolutions with applications to non-uniform pde data compression. *Journal of Computational Physics*, 498 :112636, 2024.

Kai Fukami, Taichi Nakamura, and Koji Fukagata. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Physics of Fluids*, 32 (9), 2020.

C García-Sánchez, J van Beeck, and C Gorlé. Predictive large eddy simulations for urban flows : Challenges and opportunities. *Building and Environment*, 139 :146–156, 2018.

Ioana Gavra. Méthodes de monte carlo par chaînes de markov, 2023. Cours donné à l'Université de Rennes pour les étudiants du Magistère 2 SME, durant le semestre 1.

P Gousseau, Bert Blocken, Theodore Stathopoulos, and GJF Van Heijst. Cfd simulation of near-field pollutant dispersion on a high-resolution grid : a case study by les and rans for a building group in downtown montreal. *Atmospheric environment*, 45(2) :428–438, 2011.

Anthony Gruber, Max Gunzburger, Lili Ju, and Zhu Wang. A comparison of neural network architectures for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 393 :114764, 2022a. ISSN 0045-7825. DOI: https://doi.org/10.1016/j.cma.2022.114764. URL `https://www.sciencedirect.com/science/article/pii/S004578252200113X`.

Anthony Gruber, Max Gunzburger, Lili Ju, and Zhu Wang. A comparison of neural network architectures for data-driven reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 393 :114764, 2022b.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning : data mining, inference, and prediction*, volume 2. Springer, 2009. DOI: 10.1007/978-0-387-21606-5.

Claire E Heaney, Yuling Li, Omar K Matar, and Christopher C Pain. Applying convolutional neural networks to data on unstructured meshes with space-filling curves. *Neural Networks*, 175 :106198, 2024.

Nicholas S Holmes and Lidia Morawska. A review of dispersion modelling and its application to the dispersion of particles : An overview of different dispersion models available. *Atmospheric environment*, 40(30) :5902–5928, 2006.

Juš Kocijan, Nadja Hvala, Matija Perne, Primož Mlakar, Boštjan Grašič, and Marija Zlata Božnar. Surrogate modelling for the forecast of seveso-type atmospheric pollutant dispersion. *Stochastic Environmental Research and Risk Assessment*, 37(1) :275–290, 2023. URL `https://www.researchgate.net/publication/360665374_Surrogate_modelling_for_the_forecast_of_Seveso-type_atmospheric_pollutant_dispersion`.

Eliott Lumet. *Assessing and reducing uncertainty in large-eddy simulation for microscale atmospheric dispersion.* PhD thesis, Université Toulouse III - Paul Sabatier, 2024. URL `https://theses.fr/2024TLSES003`. Accessed : 2024-05-30.

Eliott Lumet, Thomas Jaravel, Mélanie C. Rochoux, Olivier Vermorel, and Simon Lacroix. Assessing the Internal Variability of Large-Eddy Simulations for Microscale Pollutant Dispersion Prediction in an Idealized Urban Environment. *Boundary-Layer Meteorology*, 190(2) :9, Jan 2024. ISSN 1573-1472. DOI: 10.1007/s10546-023-00853-7.

Amandine Marrel, Nadia Perot, and Clémentine Mottet. Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators. *Stochastic Environmental Research and Risk Assessment*, 29(3) :959–974, Mar 2015. ISSN 1436-3259. DOI: 10.1007/s00477-014-0927-y.

Shahin Masoumi-Verki, Fariborz Haghighat, and Ursula Eicker. A review of advances towards efficient reduced-order models (rom) for predicting urban airflow and pollutant dispersion. *Building and Environment*, 216 :108966, 2022.

James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458) :415–446, 1909.

Anouar Meynaoui. Apprentissage non supervisé, 2024. Cours donné à l'Université Rennes 2 pour les étudiants de Master 1 MAS, durant le semestre 2.

Takaaki Murata, Kai Fukami, and Koji Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882 :A13, 2020. DOI: 10.1017/jfm.2019.822.

Giap Nguyen. *Courbes remplissant l'espace et leur application en traitement d'images.* Theses, Université de La Rochelle, November 2013. URL `https://theses.hal.science/tel-01174960`.

B. X. Nony, M. C. Rochoux, T. Jaravel, and D. Lucor. Reduced-order modeling for parameterized large-eddy simulations of atmospheric pollutant dispersion. *Stoch. Environ. Res. Risk Assess.*, 37(6) :2117–2144, Jun 2023. ISSN 1436-3259. DOI: 10.1007/s00477-023-02383-7.

Bastien X. Nony. *Reduced-order models under uncertainties for microscale atmospheric pollutant dispersion in urban areas : exploring learning algorithms for high-fidelity model emulation.* Phd thesis, Université de Toulouse, France, 2023.

Jean-François Paiement. Généralisation d'algorithmes de réduction de dimension. 2004. URL `https://hdl.handle.net/1866/14539`.

Carl Edward Rasmussen, Christopher KI Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006. DOI: 10.7551/mitpress/3206.001.0001.

M.C. Rochoux, E. Lumet, L. Thouron, G. Rea, F. Auguste, T. Jaravel, and O. Vermorel. Large-eddy simulation multi-model comparison of the MUST trial 2681829. Technical Report TR-CMGC-21-72, CERFACS, Toulouse, France, 2021.

Gilbert Saporta. *Probabilités, analyse des données et statistique.* Editions technip, 2006.

Michael Schatzmann, Helge Olesen, and Jörg Franke. COST 732 model evaluation case studies : approach and results. Technical report, University of Hamburg, Meteorological Institute, 2010. URL `https://www.researchgate.net/profile/George-Efthimiou-3/post/Has-fluent-been-compared-to-starccm/attachment/59d6585379197b80779ae4bd/AS%3A538043318628353%401505290931380/download/5th_Docu_May_10.pdf`. Accessed : 2023-12-01.

Lawrence Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics*, 45(3) :561–571, 1987. DOI: 10.1090/qam/910462.

Ricardo Vinuesa, Hossein Azizpour, Iolanda Leite, Madeline Balaam, Virginia Dignum, Sami Domisch, Anna Felländer, Simone Daniela Langhans, Max Tegmark, and Francesco Fuso Nerini. The role of artificial intelligence in achieving the sustainable development goals. *Nature Communications*, 11(1) :233, Jan 2020. ISSN 2041-1723. DOI: 10.1038/s41467-019-14108-y.

Songlin Xiang, Xiangwen Fu, Jingcheng Zhou, Yuqing Wang, Yizhou Zhang, Xiurong Hu, Jiayu Xu, Huazhen Liu, Junfeng Liu, Jianmin Ma, and Shu Tao. Non-intrusive reduced order model of urban airflow with dynamic boundary conditions. *Building and Environment*, 187 :107397, 2021. ISSN 0360-1323. DOI: 10.1016/j.buildenv.2020.107397.

D. Xiao, C.E. Heaney, F. Fang, L. Mottet, R. Hu, D.A. Bistrian, E. Aristodemou, I.M. Navon, and C.C. Pain. A domain decomposition non-intrusive reduced order model for turbulent flows. *Computers & Fluids*, 182 :15–27, 2019. ISSN 0045-7930. DOI: 10.1016/j.compfluid.2019.02.012.

Eugene Yee and Christopher A Biltoft. Concentration fluctuation measurements in a plume dispersing through a regular array of obstacles. *Boundary-Layer Meteorology*, 111(3) :363–415, 2004. DOI: 10.1023/B:BOUN.0000016496.83909.ee.

Résumé

Les accidents industriels survenus ces dernières années ont démontré leur impact significatif sur la santé des populations avoisinantes. La modélisation de ces phénomènes représente une avancée cruciale dans la gestion des risques sanitaires qui en découlent. Durant mon stage, j'ai étudié le modèle réduit POD—GPR qui permet d'émuler, en quelques secondes seulement, la surface a surface de réponse d'un modèle LES de mécanique des fluides avec une précision notable. L'objectif de mon stage a été de proposer une version plus performante de l'étape de réduction de dimension du modèle POD–GPR. Cette amélioration à été réalisée en optimisant l'étape de réduction de dimension grâce à l'utilisation de techniques d'apprentissage sophistiquées comme l'analyse en composantes principales à noyau (KPCA), la régression des moindres carrés partiels (PLS) et les auto-encodeurs convolutifs (CAE). J'ai démontré la pertinence du KPCA comme méthode de réduction de dimension pour les champs physiques, tout en vérifiant la possibilité d'apprentissage d'une coupe 2-D vers un champ 3D par transfert pour l'optimisation de ses hyperparamètres. Les résultats obtenues par la KPCA montrent que l'utilisation de méthodes de réduction non-linéaires pour réduire la dimension de champs physique améliore considérablement les résultats par rapport à la PCA classique. Cependant, il est encore possible d'améliorer les résultats des CAE en utilisant des méthodes par graphes.