



HAL
open science

Modélisation du tenseur de contraintes sous-maîles par réseau de neurones à convolutions 3D en Turbulence homogène isotrope

Nathaniel Saura

► **To cite this version:**

Nathaniel Saura. Modélisation du tenseur de contraintes sous-maîles par réseau de neurones à convolutions 3D en Turbulence homogène isotrope. Dynamique des Fluides [physics.flu-dyn]. Université de Lille, 2021. Français. NNT : 2021LILUI046 . tel-03482852

HAL Id: tel-03482852

<https://theses.hal.science/tel-03482852v1>

Submitted on 16 Dec 2021

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

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

UNIVERSITÉ DE LILLE

École doctorale : EDSPI UNIVERSITÉ LILLE NORD-DE-FRANCE

Département : LABORATOIRE DE MÉCANIQUE DES FLUIDES DE LILLE KAMPÉ DE
FÉRIET

Thèse de doctorat spécialité Mécanique intitulée

Modélisation du tenseur de contraintes sous-maïlles par réseau de neurones à convolutions 3D en Turbulence homogène isotrope

Présentée par **Nathaniel SAURA**
Dirigée par **Thomas GOMEZ**

Rapporteurs :

Paola CINNELLA
Professeure à la Sorbonne Université, Paris

Marcello MELDI
Assistant professeur à l'ENSMA, Université de Poitiers, Futuroscope
Chasseneuil

Examineurs :

Antoine DAZIN
Professeur à l'ENSAM, Lille **Président du jury**

Michael BAUDOIN
Professeur au IEMN, Université de Lille

Christophe FRIESS
Maître de conférences, Aix-Marseille Université

Invité :

Thomas GOMEZ
Professeur à l'Université de Lille

31 août 2021

Remerciements

Pour paraphraser le Talmud : l'Homme est un petit monde. Comme tout système dynamique, il est soumis à l'action irréductible du temps, du temps qui passe. À l'image des quatre saisons qui se succèdent, ces quatre années de thèse ont été marquées par des moments d'inspirations et d'autres de doutes, les derniers entraînant souvent les remises en questions nécessaires aux nouvelles idées, à cette vision élargie et transcendante qu'on cherche tous.

Dans le processus de dépassement de soi et de transition de l'étudiant au chercheur, chaque choix professionnel comme personnel participe aux fondations de la vie future. Je voudrais alors profiter de ces quelques lignes pour remercier mes parents qui ont toujours su trouver les mots pour parfois me faire relativiser, parfois me secouer mais toujours avec bienveillance et confiance. Je les remercie également pour avoir su me donner les clés pour m'adapter aux situations radicalement nouvelles que j'ai rencontrées, au travers leur éducation et les valeurs qu'ils ont su véhiculer et transmettre. C'est en premier lieu grâce à eux que j'ai pu faire les choix dont je suis le plus fier.

Je remercie également mes grands parents. Je me rappelle sa joie quand j'annonçais à mon grand père maternel que je m'orientais vers des études centrées sur la mécanique ; je décelais également un mélange de fierté et de nostalgie lorsque je parlais de mes cours de dessin industriel à mon grand père paternel, et que je l'écoutais me raconter ses prouesses. Je leur dédie entièrement mon étude ainsi que mes recherches. Et de même que je porte leur prénom, je les associe à ma thèse, à mes accomplissements futurs, pour toujours, au delà des limites du temps. Je remercie également mes grands mères, que j'ai la chance d'avoir physiquement près de moi, pour leur amour inconditionnel qui me marque chaque jour un peu plus, leur disponibilité, leur force. Exemples de résilience malgré des moments très difficiles, vous êtes les six lumières qui éclaireront à jamais mon existence.

Je remercie également ma sœur Annael et son mari Benny toujours prompts à passer un bon moment, à discuter de choses légères comme de choses importantes tout en s'occupant de leurs trois enfants remplis de vie et de surprises ; mon frère Ouriel qui a toujours été un modèle pour moi, de façon d'être, puis de rigueur et d'organisation, de réussite. Ces années à Lille nous ont offerts de bons souvenirs, des liens renforcés et pérennes peu importe nos destinations. Enfin, ma petite sœur Yael qui est unanimement la douceur incarnée, avec qui nous pouvons discuter des heures et qui me surprend de jour en jour. De grandes choses sont en route !

Je remercie également mon oncle Damian et ma tante Isabelle, mes cousins Ruben, Raphael et Kelly, sans oublier Eve et la petite chouchou Bluma, pour leur gentillesse, leurs coups de main

et tous ces moments passés ensemble qui ont bercés mes 20 premières années et m'emplissent de bons souvenirs.

Je remercie et pense à tata Soso, Simy, Esther, ma cousine Emily, à tonton Alain, Salomon, Itzhak, ses frères et sœurs, Mikael, Laura, Eva, Gary, tonton Marcel, Sandrine, Mélanie, Ruben, Jonas, Samuel, Caroline, leurs enfants, Sarah et ses enfants, et tous mes autres proches.

Mon séjour à Lille a été marqué par des rencontres. J'y ai découvert une approche de la vie qui m'a profondément impactée. Je ne remercierais jamais assez la communauté Juive de Lille si accueillante avec à sa tête R. E. Dahan et son épouse R. A. Dahan envoyés et vecteurs de principes, d'un espoir qui ont résonnés en moi durant ces quatre années, et pour la vie. L'apothéose de cette rencontre se retrouve en la personne d'Esther qui personnifie cet espoir, ces valeurs et rayonne d'un feu plus grand chaque jour ; elle est le présent éternel qui lie, et j'espère liera, le passé et l'avenir. Je serai toujours redevable à cette communauté pour tout ce qu'elle m'a apporté et j'espère pouvoir lui donner autant que j'ai reçu.

Le travail de thèse développé dans ce manuscrit a été mené avec mon directeur de thèse Thomas Gomez que je remercie pour, sa bienveillance, ses connaissances et bien sûr l'opportunité offerte de travailler sur un sujet aussi vaste, complexe et aux débouchés aussi variés. Également pour nos discussions riches, pleines d'idées novatrices, de projets, de détails sur la Physique, des suggestions. J'ai en mémoire des après midi entiers à réfléchir devant le tableau blanc avec nos cafés. J'espère sincèrement que cette thèse débouchera sur d'autres collaborations et que nous garderons notre entente dépassant le cadre professionnel.

Je remercie également le directeur du laboratoire de mécanique des fluides de Lille Jean Philippe Laval ainsi que Jean Marc Foucault, les doctorants et co-bureau pendant plus de trois ans pour certains : Florian (Ping à toi), Rakesh toujours prompt à rendre un service, Hussein, Tarek, Adam, Ilkay et d'autres.

Je remercie également les membres du Jury.

Cette thèse est la continuité de cinq années d'études. Durant ces années, j'y ai rencontré des personnes qui ont un rôle à part entière dans cette aventure. Je voudrais remercier Bruno Gilles qui m'a accompagné durant mon cursus d'ingénieur à l'école Polytechnique Universitaire de Lyon (Polytech Lyon) et Bastien Di Pierro qui a su me réconcilier avec la programmation, puis me conseiller et m'encourager lors du Master, des stages de recherches et de la thèse quand la situation était délicate.

Un tournant dans ma vie a été la rencontre de Louis Gostiaux au sein du Laboratoire de Mécanique des Fluides et d'Acoustique (LMFA) de Lyon au cours de mes stages de recherche. Jamais avare en encouragement, il a su réveiller en moi le goût de la recherche. J'ai en tête son enthousiasme pour ce qui était mon tout premier résultat de simulation numérique. Il a su me donner la confiance nécessaire pour aller plus loin et tout donner aux moments où il le fallait. Encore aujourd'hui, je pense à son enthousiasme et je regagne en motivation pour aller au fond de mes idées. Ces trois rencontres avec celle de Diane Micard, qui encadrait également mon premier stage, sont pour moi les plaques tournantes qui m'ont ouvert les portes de la thèse et de tout ce qui en découlera, je vous en remercie.

Pour remonter encore chronologiquement, je voudrais remercier mes amis de toujours Ishoua et Nathan. Si la vie nous a fait nous éloigner géographiquement, je sais que je pourrais toujours compter sur vous et vous savez que la réciproque est vraie. Je ne pourrai pas décrire le nombre de bons souvenirs dont vous êtes les acteurs principaux et l'impact de chacun de ces moments sur mon bien être.

De même, je voudrais remercier mes amis Lévy et Rivka Dahan ainsi que toute la famille Dahan, Menahem, Thomas, David, Jonathan, Yonathan, Yona, Nathan, David, chacun sait l'amitié que je lui porte et j'espère que nous ne cesserons pas d'être en contact.

*À mon associé, à Jojo l'ourson, vous nous manquez terriblement,
À tous ceux qui m'ont soutenu et éclairé,
À Elle,
À Toi.*

*“Pourquoi ne pas être à la hauteur des attentes que vous avez de vous même ? Soyez
l'incroyable personne que vous pensez être !”
R. M. M. Schneerson*

Résumé

La modélisation numérique de la turbulence est une des approches classiques pour étudier la complexité de la dynamique des échelles actives dans un écoulement turbulent. La richesse des mécanismes œuvrant dans ces écoulements rend quasiment impossible la prise en compte de l'ensemble des phénomènes inter-échelles. Des méthodes numériques simplifiées ne résolvant que la dynamique aux grandes échelles sont proposées ; l'influence des petites échelles sur la dynamique des grandes échelles étant introduite par une modélisation physique "sous mailles". Ces modèles numériques se sont complexifiés et améliorés parallèlement aux progrès de l'algorithme, des capacités des ordinateurs et de la compréhension des phénomènes physiques décrits par les équations de Navier-Stokes. C'est dans ce contexte qu'est né le développement de méthodes numériques couplées avec l'intelligence artificielle (IA) afin de tenir compte de manière plus fiable de la physique non résolue. Les travaux développés dans ce manuscrit apportent à la communauté de la mécanique des fluides un corpus de définitions et de méthodes permettant une exploitation plus optimale du potentiel des IA en adéquation avec la pluralité et la richesse des échelles que les écoulements turbulents exhibent. Nous discutons des composants principaux des réseaux de neurones tels que les non-linéarités induites par les fonctions d'activation, les fonctions de coût ainsi que les algorithmes d'optimisation des poids du réseau. Les propriétés, les attentes et les limites de ces hyperparamètres sont étudiées pour les optimiser. Nous avons ainsi combiné l'activation Mish avec une fonction coût multi-échelle tout en optimisant les poids du réseau avec l'algorithme Nadam. Les réseaux à convolutions profonds nécessitent une construction minutieuse. À ce titre, nous apportons les éléments nécessaires pour comprendre la complexité de l'apprentissage. Nous traitons de l'abstraction des patterns identifiés, à la propagation de l'information dans le réseau, en passant par d'autres concepts. Cette conversation s'articule autour des particularités de l'architecture U-net utilisée dans ce manuscrit. Ces travaux montrent qu'il est possible d'apprendre la relation entre le champ de vitesse 3D filtré brute, et les composantes du tenseur de stress décrivant les interactions entre échelles résolues et non résolues. C'est aussi une des solutions les plus précises proposée à ce jour. En effet, que nous regardions la prédiction ou les grandeurs dérivées - sa divergence et sa contraction avec le champ de déformations - il apparaît que les corrélations de ces grandeurs avec leurs homologues théoriques oscillent entre 90 et 99% surclassant le modèle de Clark. Il s'avère également que les performances ne sont que très peu perturbées par une augmentation de 50% ou de quasiment 200% du Reynolds. Afin d'affiner l'analyse vers l'interprétation physique des résultats, nous décomposons le tenseur prédit en trois tenseurs conformément à l'idée Leonard. Les interactions entre les échelles résolues et non résolues portées par les deux premiers termes sont mieux appréhendées par le réseau de neurones que par la modélisation de Clark. Le tenseur prédit montre une meilleure appréhension des interactions entre les échelles non résolues que le modèle de Clark ne le permet. Cela confirme que l'IA a construit une correspondance physiquement viable entre champ de vitesse filtré et tenseur de stress. Au fil de ce manuscrit, nous tentons d'apporter un point de vue critique aussi bien sur les méthodes que sur les résultats, leurs implications et leurs étendues. Le modèle hybride présenté s'inscrit dans un domaine en pleine expansion autour de la mécanique des fluides. Il élargit l'application de la modélisation numérique de la Turbulence au-delà de notre compréhension, intuition de la Physique, ou encore des capacités des ordinateurs actuels en construisant des modèles numériques plus rapides et plus fiables.

Abstract

Numerical modeling of Turbulence is one of the classical approaches for studying active scales dynamic in a turbulent flow. The interactions between the different scales of motion featured by any turbulent flow is currently mathematically and numerically intractable. Instead numerical reduced models have been proposed : large-scale motions are resolved using the large eddy equations whereas small-scale influence is modeled and injected into the large-scale dynamic. Numerical modeling is brought to grow in complexity along the progress made in connected fields (such as Algorithmics), the computer power increase and theoretical understanding of the physical phenomena lying beneath the Navier-Stokes equations. This background leads to the coupling of computational fluid mechanics (CFD) and to methods based on artificial intelligence (AI) aiming to model non solved scales impacts on large scale motions. This work brings to the CFD community a corpus of definitions and methods promoting a more optimal use of the AI potential on one hand. On the other hand, we provide a framework allowing to deal with the variety of scales. We discuss of the neural network main components such as nonlinearities induced by activation function, cost functions as well as network's weights optimization algorithms. Properties, expectations and limitations of these hyper-parameters are studied. Based on that work, we combine the Mish activation function with a custom multi-scale cost function with the Nadam network optimization algorithm. Deep convolutional neural network models have to be cautiously designed. We therefore provide elements to understand the complexity of the learning task. We go from pattern abstractions to information flow, dealing with vanishing gradients and so on. This discussion is based on the U-net architecture we used. Our works highlight the fact that learning the stress tensor that describes the non resolved - resolved/non resolved interactions from the almost raw 3D filtered velocity field is possible and is one the most accurate solutions that's been proposed to date. This is true both on the prediction itself and on the derived quantities - its divergence and its contraction with the filtered strain rate tensor. The AI based quantities correlations with their expected counterparts oscillates between 90 and 99% outperforming the Clark model. This performance isn't bound to the simulation used for the learning nor to its Reynolds ; we show that the machine learning based model accuracy doesn't seem to be affected by the increase on the Reynolds number of 50 or almost 200%. To perform a deeper analysis and try to "physically" interpret results, we perform Leonard's decomposition to identify three parts in the stress tensor. The interactions between resolved and non resolved scales carried by the two first terms are better apprehended by the neural network than the Clark model ; the predicted tensor captures much more information about interactions between non resolved scales than the algebraic model. This confirms that the machine learning model found a physics compatible mapping between filtered velocity field and stress tensor. Throughout the manuscript, we tried to give a critical point of view on the methods and their results, their involvements or their validity. This coupling widens CFD scope and proposes to overcome the limitation of our current understanding of the physics of current computer capacity to construct faster and more reliable numerical models.

Table des matières

1	Introduction	20
2	Méthodes LES pour la Turbulence ; grandeurs et outils pour l'analyse des modèles	29
2.1	Équations constitutives de la mécanique des fluides	29
2.2	Phénoménologie de la turbulence	33
2.2.1	Le nombre de Reynolds et hypothèses de Kolmogorov	33
2.2.2	Corrélation et analyse spectrale de l'écoulement	35
2.3	Modélisation des grandes échelles et méthodes LES	40
2.3.1	Filtrage et problème de fermeture	40
2.3.2	Modèles de Smagorinsky et du Gradient	43
2.4	Évaluation des modèles : outils d'analyse	45
2.4.1	Grandeurs à évaluer	45
2.4.2	Outils pour évaluer la qualité des champs modélisés	47
3	Apprentissage automatique par réseaux de neurones artificiels	52

3.1	Introduction aux réseaux de neurones	52
3.1.1	Fonctionnement et apprentissage d'un réseau de neurones	52
3.1.2	Les réseaux de neurones à convolutions	57
3.2	Activations saturées ou sporadiques : fonctions d'activations	61
3.3	Objectif et alignement : fonction de coût, régularisation et méthodes d'optimisation	65
3.3.1	La fonction de coût et les métriques	65
3.3.2	Régularisation sur les paramètres du réseau	67
3.3.3	Algorithmes d'optimisation : du SGD au Nadam	69
4	Élaboration d'une intelligence artificielle pour la mécanique des fluides	76
4.1	Historique de l'évolution du couplage Mécanique des Fluides et réseau de neurones	76
4.1.1	Injection de l'IA dans l'équation de fermeture des modèles RANS	76
4.1.2	Prédiction de la partie anisotropique du tenseur de Stress (RANS) et injection d'invariance	78
4.1.3	Travaux effectués dans le contexte de la LES	81
4.2	Cadre de travail : ensemble d'entrée-sorties et réseau U-net	83
4.2.1	Élaboration des bases de données	83
4.2.2	L'architecture U-net et ses propriétés	89
4.3	Sélection et impact des hyper-paramètres	94
4.3.1	Paramètres fixés en amont	94
4.3.2	Illustration de l'impact de certains hyper-paramètres	96
4.4	Évaluation du modèle sélectionné sur les ensembles d'apprentissage et de test . .	101

5	Généralisation de la modélisation du tenseur des contraintes sous mailles par réseaux à convolutions 3D	108
5.1	Étude des performances du modèle augmenté pour une simulation DNS dont $Re_\lambda \approx 140$	109
5.1.1	Impact de la résolution sur les prédictions en régime développé	112
5.1.2	Prédiction à l'entrée du régime développé : étude des composantes de la décomposition de Leonard	120
5.1.3	Capacité d'adaptation par rapport à des champs turbulents de nature fondamentalement différente	127
5.2	Prédictions sur une simulation avec $Re_\lambda \approx 240$	134
5.2.1	Analyse des performances de la prédiction dans un écoulement plus turbulent dans son régime développé	136
5.2.2	Prédiction dans le régime dissipatif : étude des performances et des transferts d'énergie à travers la décomposition de Leonard	143
5.2.3	Prédiction dans le régime productif : étude des performances et des transferts d'énergie à travers la décomposition de Leonard	151
6	Conclusions des travaux et thèmes d'ouverture	161
A	Suppléments sur la minimisation d'une fonction de coût et le moment	174
A.1	Limite de la descente de gradient basique	174
A.2	Utilisation du moment en substitution à la Hessienne : moment classique et de Nesterov	176

B Plus de détails sur les résultats présentés Chapitre 5	179
B.1 Résultats complémentaires pour la simulation dont Reynolds $Re_\lambda = 140$	179
B.1.1 Figures supplémentaires pour l'itération 118	179
B.1.2 Figures supplémentaires pour l'itération 88	181
B.1.3 Figures supplémentaires pour l'itération 42	183
B.1.4 Figures et scores pour l'itération 25	184
B.2 Résultats complémentaires pour la simulation dont Reynolds $Re_\lambda = 240$	185
B.2.1 Sur l'itération 125	185
B.2.2 Sur l'itération 61	187

Table des figures

1.1	Différents niveaux de description et les formalismes établis pour les étudier (Mat-tila, 2010).	21
1.2	Peinture de Léonard de Vinci illustrant la turbulence dans un fluide.	22
1.3	Spectre d'énergie cinétique turbulente (croix jaunes) et identification des gammes d'échelles : grandes échelles, inertielle et dissipative. Une courbe (pointillés rouges) de pente $k^{-5/3}$ est tracée en référence selon la théorie de Kolmogorov.	23
1.4	Exemples des intrications des structures tourbillonnaires dans un champs turbulent. En bleu iso 6 ; en jaune iso -8 ; en rouge iso -11. Le Reynolds à l'échelle de Taylor (voir corps) est $Re_\lambda = 90$ et la résolution est de 128 points de discrétisation.	24
1.5	Évolution temporelle de la Turbulence au sein d'un écoulement.	28
2.1	Deux dynamiques au sein du champ de vitesse. La courbe noire représente le champ de vitesse $U_x(x, 0, 0)$; la courbe bleue représente cette même grandeur dont les hautes fréquences ont été coupées.	35
2.2	Illustrations des interactions triadiques des vecteurs \mathbf{k} , \mathbf{p} et \mathbf{q} vérifiant $\mathbf{k} = \mathbf{p} + \mathbf{q}$	38
2.3	Spectre d'énergie cinétique turbulente et identification des grandes échelles, et des gammes inertielle et dissipative. Une courbe de pente $k^{-5/3}$ est tracée en référence selon la théorie de Kolmogorov.	39
2.4	Équivalence et interprétation physique (gauche) et spectrale (droite) du filtrage des grandeurs. Dans la figure de gauche les ovales foncées représentent les quantités (e.g. la vitesse) non résolues nécessitant un modèle. Idem pour la région hachurée figure de droite en terme de nombre d'onde et de spectre d'énergie (Vollant, 2015).	40
2.5	Visualisation de l'effet du filtrage dans l'espace de Fourier. Les petites structures correspondant aux nombres d'onde supérieur à κ_c sont absentes après filtrage Fig.(2.5b).	41
2.6	Pour une simulation dont le Reynolds à l'échelle de Taylor est $Re_\lambda \approx 240$, nous comparons la première composante des trois tenseurs de Eq.(2.3.6). Les courbes brisées rouge et orange représentent la modélisation de ces grandeurs par le modèle de Clark, et de Smagorinsky. Les courbes noires représentent les grandeurs issues de la DNS filtrée.	47

2.7	Pour les trois graphiques, la courbe brisée rouge et orange représentent les modélisations de ces grandeurs par le modèle de Clark, et Smagorinsky (respectivement). Les courbes noires représentent les grandeurs issues de la DNS filtrée.	48
3.1	Définition d'un neurone : unité de réinterprétation de l'information entrante permettant au réseau de la résumer et de la moduler dans un but plus global <i>d'apprentissage</i> . Les poids synaptiques, ainsi que le biais propre au neurone constituent ce que nous avons appelés les paramètres du réseaux (Haykin et al., 2009).	53
3.2	Représentation de la convolution d'un filtre de taille 5x5 avec une entrée composée de trois matrices de taille 32x32. La carte d'activation est formée à partir de cette convolution sur toute l'entrée. On note que le filtre possède une dimension supplémentaire égale au nombre de matrices en entrée; la convolution se fait simultanément sur toutes les composantes de l'entrée. Image issue de http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture05.pdf	59
3.3	Propagation de l'information selon les connections entre couches (Finlayson, 2020). Les ANN considèrent que les entrées de chaque couche sont indépendantes des entrées des couches précédentes. Au contraire le CNN construit des couches de patterns plus complexes les unes que les autres et conservent une forme de mémoire des entrées des couches depuis l'entrée globale. Les Recurrent Neural Networks (RNN) (figures du bas) cherchent des liens dynamiques entre les époques d'entraînement en faisant circuler les informations des époques précédentes. Cela peut se faire séquentiellement ou de manière plus spécifiée.	59
3.4	Ci contre - tracés des fonctions tangente hyperbolique et sigmoïde.	62
3.5	Graphiques de différentes fonctions d'activation, des plus classiques comme la Sigmoïde (ligne violette) jusqu'aux plus exotiques comme l'activation Mish (ligne cyan).	63
3.6	Visualisation des normes L_1 et L_2 dans la recherche des poids optimaux (Mohri et al., 2018).	68
3.7	Composition des gradient et moment pour l'établissement de la direction à suivre pour la minimisation de \mathcal{J} dans la méthode du moment classique.	70
4.1	TBNN proposé par Ling et al. (2016b)	80
4.2	La courbe 2 illustre la relation entre la quantité d'énergie transférée à travers la coupure et les modes associés. La grandeur k_m est le nombre d'onde de coupure considéré dans l'opération de filtrage, que nous avons noté κ_C . Les deux autres courbes dépassent le cadre de ce manuscrit. Figure issue de Kraichnan (1976).	85
4.3	Spectre initial du champ de vitesse	87

4.4 Évolutions temporelles des grandeurs turbulentes (croix noires) dans la simulation utilisée pour la construction des données d'apprentissage. Les croix rouges représentent les itérations des instantanées utilisées pour construire l'ensemble d'apprentissage ; les cercles jaunes représentent celles utilisées pour l'ensemble de test (ou validation). 89

4.5 Réseau de neurones à convolutions U-net tel qu'il a été utilisé dans ces travaux. Le nombre de convolutions par étages (représenté par les blocs bleus voir légende (4.5b)) varie dans certains cas. Ce paramètre est appelé Noc par la suite pour *Number of convolutions* voir notamment Tab.(4.2). 90

4.6 Connexion résiduelle et propagation alternative de l'information He et al. (2015) 93

4.7 Comparaisons des PDFs des prédictions issues des CNN qui diffèrent entre eux de par leur fonction d'activation. Les différentes lignes brisées représentent ces résultats, la ligne noire constitue la distribution de τ_{xx} et τ_{xy} théorique respective. 98

4.8 Évolutions des fonctions de coût et métriques en fonction des époques globales d'apprentissage. Les cercles représentent des évaluations de ces fonctions sur l'ensemble d'apprentissage alors que les triangles inversés traduisent leurs scores sur l'ensemble de test 101

4.9 Diagramme de dispersion pour τ_{ij}^{ML} 102

4.10 Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales ; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales. 104

4.11 Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales ; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales. 107

4.12 Diagramme de dispersion pour τ_{ij}^{ML} 107

5.1 Évolution temporelle des grandeurs caractéristiques de la Turbulence pour la simulation dont la résolution est $N = 176$ et $Re_\lambda \approx 140$ dans le régime développé. 110

5.2 Ci-contre - Spectre de l'énergie cinétique calculée dans le régime développé de la turbulence au sein de l'écoulement issue de la simulation au Reynolds $Re_\lambda \approx 140$. 110

5.3 Illustrations des premiers et derniers u_x balayées lors de l'apprentissage (figures du haut) et lors des tests de cette section de (figures du bas) Voir tableua Tab.(5.1). 111

5.4 Corrélations de Pearson entre les composantes de τ_{ij}^{FDNS} et celles de τ_{ij}^{ML} (figures à gauche). Idem avec les composantes de τ_{ij}^{Clark} (figures à droite) à la place de la prédiction. 114

5.5 À l'itération 118, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 116

5.6 **Figures du haut** : Coupe de la composante τ_{xy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Smagorinsky (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Smagorinsky et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$. Les échelles sont volontairement propres à chacune des coupes. 117

5.7 À l'itération 118, comparaisons des PDFs des composantes du vecteur $\partial_{x_j}\tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenues à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte) ; Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 118

5.8 **Figures du haut** : Coupe de la composante $(\nabla \cdot \tau)_z$ FDNS (à gauche), prédite par IA (au centre), et obtenue par le modèle de Clark (à droite). Les échelles de couleurs sont celles de la FDNS. **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$, à l'itération 118. 119

5.9 À l'itération 88, comparaisons des PDFs de L_{yy} et L_{yz} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de L_{ij}^{FDNS} 122

5.10 À l'itération 88, comparaisons des PDFs de C_{xx} et C_{xy} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de C_{ij}^{FDNS} 122

5.11 À l'itération 88, comparaisons des PDFs de R_{zz} et R_{xy} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de R_{ij}^{FDNS} 123

5.12 **Figures du haut** : Coupe de la composante R_{zz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 3$ 125

5.13 **Figures du haut** : Coupe de la composante R_{xy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 3$ 126

5.14 **Figures du haut** : Coupe de la composante R_{xy}^{FDNS} (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$. Les échelles de couleurs sont celles de la FDNS. 127

5.15 À l'itération 42, comparaisons des PDFs des composantes x et z de $\partial_{x_j}\tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenues à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte); Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 129

5.16 Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) aux niveaux tensoriel, vectoriel et scalaire. 132

5.17 Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) des trois types d'interactions composants le tenseur τ conformément à la décomposition de Leonard (2.3.6). 133

5.18 Évolution temporelle des grandeurs caractéristiques de la Turbulence pour la simulation dont la résolution est $N = 256$ et $Re_\lambda \approx 240$ dans le régime développé. 134

5.19 Spectre de l'énergie cinétique calculée dans le régime développé de la turbulence au sein de l'écoulement issue de la simulation au Reynolds $Re_\lambda \approx 240$ 134

5.20 Illustrations des premiers et derniers u_x balayées lors de l'apprentissage (figures du haut) et lors des tests de cette section de (figures du bas) Voir tableau Tab.(5.1). 135

5.21 À l'itération 125, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 138

5.22 À l'itération 125, erreurs relatives entre τ_{ij}^{FDNS} et τ_{ij}^{ML} (figures de gauche). Idem entre τ_{ij}^{FDNS} et τ_{ij}^{Clark} (à droite). 140

5.23 À l'itération 125, comparaisons des PDFs des composantes x et z de $\partial_{x_j}\tau_{ij}$ ainsi que des scalaires Π et Ψ théoriques (courbes continues noires), prédites et issues des modèles Clark et de Smagorinsky (ligne brisée rouge, lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS. . . . 141

5.24 Erreurs relatives le long de l'axe x . Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales. Les écarts sont calculés à l'itération 88. 143

5.25 À l'itération 88, comparaisons des PDFs des composantes τ_{xx} et τ_{yz} , $(\nabla \cdot \tau)_y$ et $(\nabla \cdot \tau)_z$ ainsi que des scalaires Π et Ψ calculées théoriquement (courbe solide noire), prédites (ligne brisée rouge) et modélisées (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS. 145

5.26 À l'itération 88, comparaisons des PDFs de R_{yy} et R_{xz} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de R_{ij}^{FDNS} 147

5.27 **Figures du haut** : Coupe de la composante R_{yy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$ 148

5.28 **Figures du haut** : Coupe de la composante R_{xz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$ 149

5.29 À l'itération 88, comparaisons des PDFs des champs scalaires $\Pi_{\mathbf{R}}^{\text{FDNS}}$ (courbe continue noire), avec $\Pi_{\mathbf{R}}^{\text{ML}}$ (ligne brisée rouge), $\Pi_{\mathbf{R}}^{\text{Clark}}$ et $\Pi_{\mathbf{R}}^{\text{Smago}}$ (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS. 150

5.30 À l'itération 61, comparaisons des PDFs des composantes τ_{zz} et τ_{xy} , $(\nabla \cdot \tau)_x$ et $(\nabla \cdot \tau)_y$ ainsi que des scalaires Π et Ψ calculées théoriquement (courbe solide noire), prédites (ligne brisée rouge) et modélisées (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS. 153

5.31 Évolutions scores de Pearson en fonction du nombre de Reynolds. Les courbes solides sont les scores calculés en considérant la prédiction; en pointillées sont obtenues en considérant les modélisation de Clark. L'indice a symbolise les scores pour l'itération avant le pic de dissipation; t après l'établissement du régime développé de la turbulence 155

5.32 Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) aux niveaux tensoriel, vectoriel et scalaire. 157

5.33 Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) des trois types d'interactions composants le tenseur τ conformément à la décomposition de Leonard (2.3.6). 158

5.34 En bleu iso 0.8; en jaune iso 0.13; en rouge iso 0.24. Le Reynolds est $Re_\lambda = 240$ et la résolution est de 256 points de discrétisation. 159

A.1 Illustration de la divergence due à une mauvaise évaluation du pas de recherche dans l'espace des paramètres et de l'absence de terme compensant les oscillations. 175

A.2 Illustration de l'action du moment sur la direction dans l'espace des paramètres que l'optimiseur va finalement emprunter. Un cas limite apparaît lorsque la courbe verte (direction m_t) pointe dans un mauvais sens ou dans le sens opposé de la correction proposée par le gradient (en rouge g_t). 177

B.1 À l'itération 118, comparaisons des PDFs des champs scalaires Ψ^{FDNS} (courbe continue noire), avec Ψ^{ML} (ligne brisée rouge), Ψ^{Clark} et Ψ^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moyennes et déviations de la FDNS. 179

B.2 Erreurs relatives entre les composantes de τ_{ij}^{FDNS} et celles de τ_{ij}^{ML} (figures à gauche). Idem avec les composantes de τ_{ij}^{Clark} (figures à droite) à la place de la prédiction. 180

B.3 À l'itération 88, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 181

B.4 À l'itération 88, comparaisons des PDFs des composantes y et z du vecteur $\partial_{x_j}\tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenues à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte); Les PDFs sont standardisées par les moments de la FDNS. 182

B.5 À l'itération 42, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS. 183

B.6 Illustrations des champs de vitesse u_y pour l'itération 25 et 44 dans le plan $y = 3$. 184

B.7 **Figures du haut** : Coupe de la composante τ_{xx} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). Les échelles de couleurs sont celles de la FDNS. **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 0$ 186

B.8 **Figures du haut** : Coupe de la composante τ_{yz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Smagorinsky (à droite). Les échelles de couleurs sont celles de la FDNS. **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Smagorinsky et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 0$ 186

Liste des tableaux

2.1	Scores sur $\Pi_{L_{ij}}^{\text{model}}$, $\Pi_{C_{ij}}^{\text{model}}$ et $\Pi_{R_{ij}}^{\text{model}}$ pour les deux modèles.	50
4.1	Couples $N_c - R$ sur le pourcentage de transferts capturés par la dynamique contenu dans le sous-domaine cubique. Les valeurs de \tilde{L}_{\min} sont calculées utilisant la formule (4.2.3).	85
4.2	Comparatif des performances sur les ensembles d'apprentissage et de test pour des réseaux avec chacun une fonction d'activation différente. La profondeur du réseau est de $\text{NOC} = 2$. La police appuyée souligne les meilleurs performances. Le suffixe VAL désigne l'ensemble de test.	97
4.3	Comparatif des performances sur les ensembles d'apprentissage et de test pour des réseaux qui diffèrent de par leur fonction d'activation. La profondeur du réseau est de $\text{NOC} = 3$. La police appuyée souligne les meilleurs performances. Le suffixe VAL désigne l'ensemble de test.	99
4.4	Comparaisons des performances de CNN entraînés avec différentes fonctions de coût.	100
4.5	À l'itération 75, Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij}	104
4.6	À l'itération 44, Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij}	105
5.1	Caractéristiques des simulations considérées dans ce rapport.	109
5.2	Scores obtenus sur la prédiction au niveau tensoriel à l'itération 118. Les valeurs en noir et en orange représentent resp. les scores obtenus par l'IA et par le modèle Clark sur τ_{ij}	112
5.3	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 118. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes $\partial_{x_j}\tau_{ij}$ (niveau vectoriel) et les grandeurs du niveau scalaire Π et Ψ	113
5.4	Scores obtenus sur la prédiction au niveau tensoriel à l'itération 88. Les valeurs en noir et en orange représentent resp. les scores obtenus par l'IA et par le modèle Clark sur τ_{ij}	120

5.5	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j}\tau_{ij}$ et des grandeurs scalaires Π et Ψ	121
5.6	Scores obtenus sur les composantes R_{ij} calculées selon la formule Eq.(2.4.3), pour l'itération 88. Les valeurs en noir représentent les scores sur R_{ij}^{ML} construites à partir de τ_{ij}^{ML} ; en orange , les composantes R_{ij}^{Clark} calculées à partir des τ_{ij}^{Clark}	123
5.7	Scores obtenus sur les différentes dissipations reliées aux termes de la décomposition de Leonard pour l'itération 88. En noir, les scores évalués sur les grandeurs issues de τ_{ij}^{ML} ; en orange , les scores calculés en considérant les grandeurs issues de τ_{ij}^{Clark}	125
5.8	Scores obtenus sur la qualité de la prédiction au niveau tensoriel pour l'itération 42. Les valeurs en noir représentent les scores sur les prédictions de l'IA. En orange , nous renseignons à titre de comparaison la qualité de la modélisation par le modèle Clark des composantes τ_{ij}	128
5.9	Scores obtenus aux niveaux vectoriel et scalaire pour l'itération 44. Les valeurs en noir représentent les scores obtenus par les prédictions de l'IA. En orange , nous renseignons les scores globaux pour le modèle Clark des composantes $\partial_{x_j}\tau_{ij}$ et les grandeurs scalaires Π et $u_i\partial_{x_j}\tau_{ij}$	128
5.10	Scores au niveau tensoriel à l'itération 125. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark.	136
5.11	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 125. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j}\tau_{ij}$ et des grandeurs scalaires Π et Ψ	139
5.12	Scores obtenus sur la prédiction au niveau tensoriel à l'itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij}	142
5.13	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j}\tau_{ij}$ et des grandeurs scalaires Π et Ψ	144
5.14	Scores obtenus sur les composantes R_{ij} calculées selon la formule Eq.(2.4.3), pour l'itération 88. Les valeurs en noir : scores sur R_{ij}^{ML} ; en orange scores sur R_{ij}^{Clark}	146
5.15	Scores obtenus sur les dissipations reliées aux termes de la décomposition de Leonard pour l'itération 88. En noir et en orange , les scores évalués sur les grandeurs issues de τ_{ij}^{ML} et celles issues de τ_{ij}^{Clark} respectivement.	149
5.16	Pourcentages de rétrodiffusion pour Π calculé selon la formule théorique et celles relatives à chacune des composantes de la décomposition de Leonard. En noir les valeurs théoriques, en rouge et orange celles construites à partir de τ_{ij}^{ML} et τ_{ij}^{Clark} respectivement.	150

5.17	Scores obtenus sur la prédiction au niveau tensoriel à l'itération 61. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij}	152
5.18	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 61. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j}\tau_{ij}$ et des grandeurs scalaires Π et Ψ	154
B.1	Scores obtenus sur la qualité de la prédiction au niveau tensoriel pour l'itération 25. Les valeurs en noir et En orange représentent respectivement les scores sur les prédictions de l'IA et sur la modélisation (Clark) des composantes τ_{ij}	184
B.2	Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 25. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j}\tau_{ij}$ et des grandeurs scalaires Π et Ψ	185
B.3	Scores obtenus sur la qualité du tenseur L_{ij} pour l'itération 61. Les valeurs en noir et En orange représentent respectivement les scores sur les prédictions de l'IA et sur la modélisation (Clark).	187
B.4	Scores obtenus sur la qualité du tenseur C_{ij} pour l'itération 61. Les valeurs en noir et En orange représentent respectivement les scores sur les prédictions de l'IA et sur la modélisation (Clark).	187

Chapitre 1

Introduction

Que l'on regarde en soi - l'écoulement du sang dans nos veines et artères - autour de soi - les courants aériens/marins, le mouvement ainsi que la forme des nuages, des vagues, ou encore le vol des oiseaux, des avions - ou que l'on porte son regard au loin - les plasma stellaires, les nuages de gaz géants nourriciers des jeunes étoiles en formation, ou encore des jets énergétiques des corps célestes - nous constatons l'effet d'un corpus de lois de la Nature. Ces lois, l'Homme a tenté de les décrire au travers du formalisme des Mathématiques ; les équations qu'il a déduites de ses observations, expériences et raisonnements logiques forment la science appelée la Mécanique des Fluides. Le terme mécanique fait allusion à l'étude de la dynamique d'un système dans un milieu par rapport à un référentiel ; le fluide, qui n'a pas vraiment de définition établie, englobe tout milieu matériel parfaitement déformable ; les exemples les plus classiques sont les liquides, les gaz, les plasmas.

Cette science, construite au cours des siècles par des mathématiciens, physiciens, ingénieurs de renoms comme Euler, Cauchy, Navier, Stokes et d'autres, n'a pas encore délivré tous ses secrets : les équations de Navier-Stokes régissant la dynamique des fluides ont été établies entre les années 1820 et 1840 et depuis, elles résistent aux cerveaux les plus brillants, si bien que la preuve de l'existence d'une solution générale n'a toujours pas été établie, à plus forte raison son unicité. Cette même Nature constitue un domaine de recherche expérimentale sans limite, pour essayer de comprendre, de reproduire et d'adapter certains mécanismes en vue de répondre aux problématiques actuelles : par exemple comprendre les rôles des courants marins gouvernant l'équilibre thermodynamique entre l'océan et l'atmosphère et tous les phénomènes météorologiques et écologiques qui en découlent.

L'étude théorique et expérimentale de la mécanique des fluides s'appuie sur certaines hypothèses à la base de sa formulation. Cette mécanique fait partie d'un cadre plus global : celui de la mécanique des milieux continus. Pour définir un milieu continu, il faut au préalable se plonger dans la structure moléculaire d'un fluide, et le considérer à l'échelle microscopique. On verrait alors la matière telle qu'elle est : dans le cas d'un gaz, nous verrions des myriades de molécules en agitation qui, lorsqu'elles ne se collisionnent pas, sont séparées l'une de l'autre par des espaces vides dépassant très largement la taille de la molécule elle-même (on parlera de libre parcours moyen) ; dans le cas d'un liquide, les molécules étant beaucoup plus "proches", autant que les forces répulsives entre les molécules le permettent, la sensation de vide y serait moins forte, mais dans ce cas la répartition de la masse dans le volume serait fortement affectée

par la centralisation de la masse des particules au sein de leur noyau. À cette échelle, il est nécessaire de prendre en compte la répartition non-uniforme des masses et des molécules pour modéliser cette réalité, c'est l'objet de la mécanique statistique et des équations de Boltzmann. En contraste, la mécanique des milieux continus se place à l'échelle macroscopique (échelle plus grande que plusieurs fois la taille de la molécule) ne considérant que des variables observables, mesurables à cette échelle. À cette échelle donc, la nature apparaît comme un continuum, dans lequel les variables observables sont définies à chaque portion de fluide. Ces portions de fluides sont appelées parcelles de fluide et elles sont à la mécanique des fluides ce qu'est le point à la mécanique des solides. Les niveaux de description et les formalismes associés sont représentés dans la figure (1.1).

En 1883, Osborne Reynolds étudia les régimes d'écoulements des fluides. Il constate que au plus les effets inertiels - ou d'entraînements par le fluide autour de la parcelle considérée - sont dominants par rapport aux effets visqueux - dissipatifs - et au moins le fluide est capable d'absorber les fluctuations et instabilités qui pourraient apparaître dans le champ de vitesse. Il met au point un nombre sans dimension, qui porte aujourd'hui son nom, avec lequel il caractérise trois différents régimes. Pour un nombre de Reynolds mesuré inférieur à quelques milliers, l'écoulement est dit laminaire¹. Il est caractérisé par une topographie régulière. Il est stable, et ses lignes de courant pointent vers la même direction, chacune parallèle à l'autre. Pour des nombres de Reynolds plus élevés, jusque la dizaine de milliers, cette topographie devient de moins en moins régulière et des instabilités commencent à être visibles. Dans cette gamme de Reynolds, le fluide reste capable de contenir les instabilités, sans pour autant les dissiper. Ce régime est transitoire puisque généralement les instabilités sont soit absorbées par le fluide, soit se diffusent dans le reste de l'écoulement modifiant complètement l'allure de cet écoulement. À partir de la dizaine de milliers et au delà, les instabilités dans le champ de vitesse ne sont plus absorbées et génèrent des structures tourbillonnaires de plus ou moins grandes tailles ; l'apparition de ces structures est la marque du régime turbulent, ce sont ces structures que Leonard de Vinci avait identifiées et dépeignait (voir Fig.(1.2)). Le non-amortissement des instabilités rend l'écoulement turbulent sensible à toute variation dans le champ de vitesse ;

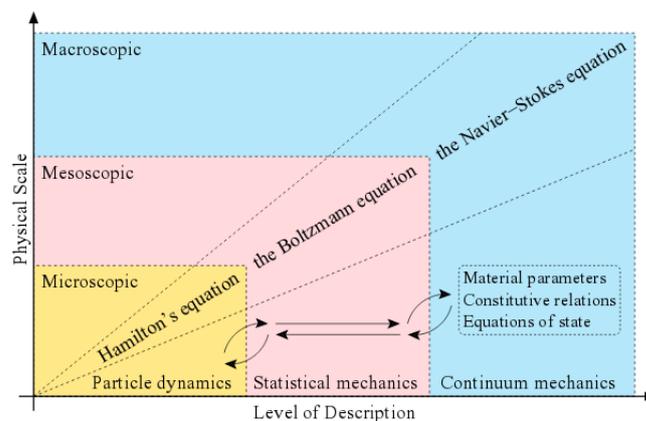


Figure 1.1 – Différents niveaux de description et les formalismes établis pour les étudier (Mattila, 2010).

1. Nous considérons le cas le plus classique

on parlera d'écoulement chaotique. Dans l'ensemble de figures Figs.(1.5), nous visualisons la déstabilisation d'un écoulement de Taylor-Green, puis la diffusion des instabilités dans l'ensemble de l'écoulement. Pour une visualisation 3D des interactions entre échelles, il est commode de représenter les iso-valeurs de vorticité aussi appelée vecteur tourbillon.

Sans rentrer dans les détails mathématiques qui seront abordés plutôt dans le corps de ce document, nous schématisons le nombre de Reynolds, noté Re par :

$$Re = \frac{\text{Effets inertiels}}{\text{Effets visqueux}}$$

Historiquement, après Reynolds c'est Richardson qui conceptualisa le premier la turbulence au sens moderne. Dans les années 1920 alors qu'il pose les bases de la météorologie moderne² il définit ce que nous appelons aujourd'hui la turbulence comme étant la composition de structures tourbillonnaires de tailles différentes, il en écrit même un poème :

*Big whorls have little whorls
that feed on their velocity, and little whorls
have smaller whorls and so on to viscosity - in the molecular sense*

Il introduit plusieurs notions fondamentales : tout d'abord le fait que les grands tourbillons contiennent (potentiellement) des plus petits tourbillons qui se concrétisent par la distribution de la vitesse (c'est-à-dire de l'énergie cinétique) de cette grande structure aux plus petites. Dans ses mots, il introduit aussi le transfert d'énergie sans perte, de structures en structures, jusqu'à la dissipation de cette énergie cinétique par des effets de frottements moléculaires caractérisant la viscosité du fluide. Enfin, à travers l'idée d'énergie cinétique portée par chaque échelle, Richardson permet d'envisager la façon la plus naturelle d'étudier la turbulence : utiliser l'espace de Fourier. Comme on le verra dans le corps du manuscrit, cet espace permet une représentation très visuelle de la cascade énergétique décrivant la cascade des échelles, théorisée par Richardson.

En 1941, Kolmogorov³ approfondit et précisa la théorie de la cascade d'énergie de Richardson. Se faisant, il posa la base de l'étude analytique, expérimentale et numérique des écoulements



Figure 1.2 – Peinture de Léonard de Vinci illustrant la turbulence dans un fluide.

2. (Richardson, 1992)

3. (Kolmogorov, 1941)

turbulents. Il établit que les contraintes géométriques et les propriétés éventuelles des milieux délimitant l'écoulement (conduite, canal, couche limite atmosphérique etc.) n'impactent que les grandes échelles. En effet, les modifications de ces *conditions aux limites* du domaine se traduisent par la génération de structures de grandes échelles, dont la taille est souvent de l'ordre de grandeur imposé par le domaine dans lequel le fluide évolue. Ces échelles se brisent en interagissant entre elles, et produisent des échelles plus petites; s'en suit le processus de cascade dans une gamme d'échelle où l'énergie véhiculée décroît de façon universelle en suivant une loi en $k \sim k^{-5/3}$. Ces échelles constituent la gamme inertielle de la cascade énergétique. Ce processus se poursuit jusqu'à l'échelle dite de Kolmogorov notée η , définie de façon universelle par le taux d'injection d'énergie cinétique aux grandes échelles (dont les causes sont extérieures au fluide), et la dissipation du fluide étudié. Ainsi, il est possible de connaître exactement l'ordre de grandeur des échelles dissipatives avant même que la cascade se mette en place. De plus, puisque le transfert d'énergie au sein de la gamme inertielle se fait sans perte, Kolmogorov et Richardson supposent que le taux de dissipation de l'énergie cinétique des échelles dissipatives, noté ε est exactement le taux d'injection de cette même énergie aux grandes échelles. En ordre de grandeur, Kolmogorov écrit

$$\eta \sim \left(\frac{\nu^3}{\varepsilon} \right)^{1/4}$$

Au delà de cette échelle, d'autres échelles sont créées mais l'énergie véhiculée décroît drastiquement voir Fig.(1.3). Depuis, beaucoup d'efforts ont été faits pour comprendre la physique de la Turbulence; malgré tout le modèle de Richardson-Kolmogorov reste la référence.

Avec l'essor de la puissance des ordinateurs disponibles en laboratoire et à plus forte raison ceux des super-ordinateurs, la simulation numérique de la physique est devenue un outil indispensable accompagnant la recherche fondamentale. On distingue deux types de simulations. Le premier propose la résolution des équations de Navier-Stokes sur toutes les échelles spatiales nécessitant une discrétisation très serrée du domaine d'étude pour s'assurer de simuler jusqu'à l'échelle de Kolmogorov. Le nombre de points de discrétisation nécessaires est de l'ordre

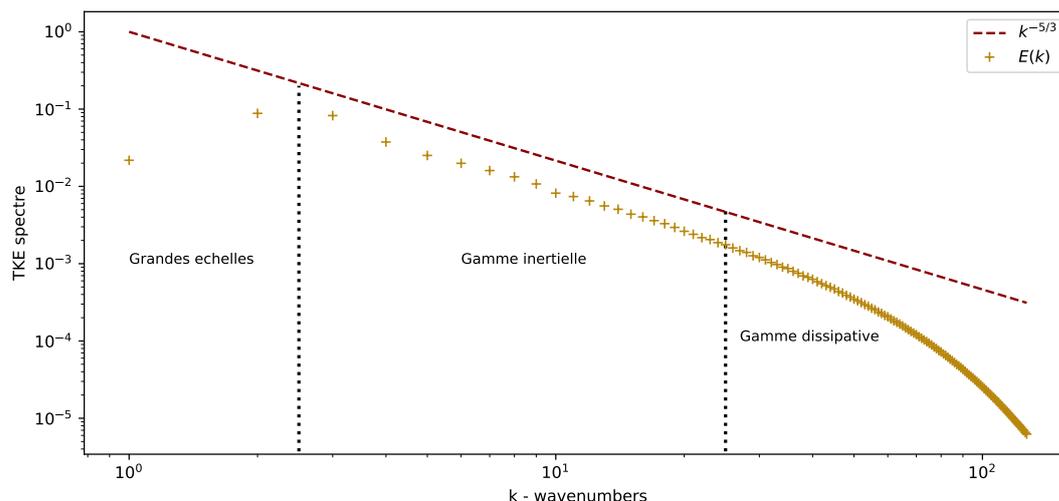


Figure 1.3 – Spectre d'énergie cinétique turbulente (croix jaunes) et identification des gammes d'échelles : grandes échelles, inertielle et dissipative. Une courbe (pointillés rouges) de pente $k^{-5/3}$ est tracée en référence selon la théorie de Kolmogorov.

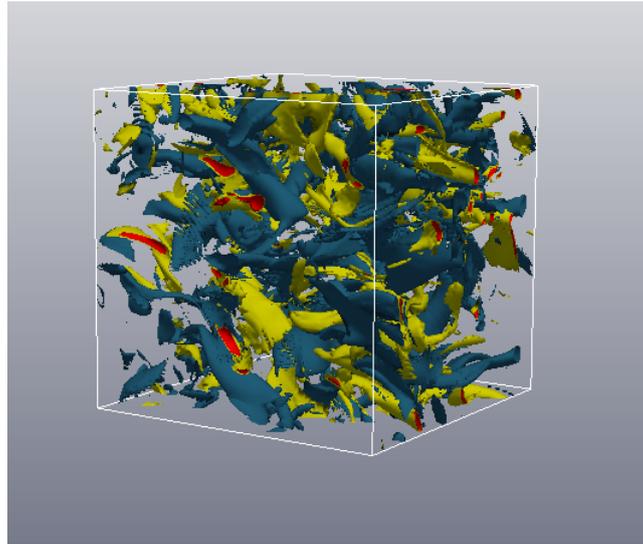


Figure 1.4 – Exemples des intrications des structures tourbillonnaires dans un champs turbulent. En bleu iso 6 ; en jaune iso -8 ; en rouge iso -11. Le Reynolds à l'échelle de Taylor (voir corps) est $Re_\lambda = 90$ et la résolution est de 128 points de discrétisation.

de $Re^{9/4}$ ⁴. Physiquement, ce nombre exorbitant s'explique par le fait qu'au plus le Reynolds est élevé, au moins le fluide peut dissiper les instabilités poussant la zone inertielle vers les échelles de très petites tailles. Les échelles dissipatives sont alors encore plus petites, et le pas de discrétisation du domaine doit être nécessairement au moins deux fois plus petit que ces échelles. On comprend que malgré les prouesses technologiques et les capacités des meilleurs ordinateurs, la simulation exacte d'un écoulement réel (dont le Reynolds est de l'ordre du million pour l'écoulement de l'air dans une zone ouverte) est limitée. C'est pourquoi les chercheurs ont mis au point un deuxième type de simulation dans lequel une partie de la Physique n'est pas résolue mais modélisée à partir de relations construites par des considérations expérimentales, empiriques etc. Les résultats seront donc moins fiables qu'avec les méthodes de simulations directes mais, en contrepartie, des simulations avec de plus hauts Reynolds peuvent converger dans des temps raisonnables. Citons deux des approches les plus couramment utilisées :

RANS : L'approche dite *moyennée* consiste à décomposer toutes les variables apparaissant dans les équations de Navier-Stokes en une somme d'une composante moyenne et de la fluctuation autour de sa moyenne. Ainsi ces procédures résolvent les équations de Navier-Stokes moyennées ou Reynolds Averaged Navier-Stokes, et *modélisent* la quantité de mouvement que le champ fluctuant injecte dans le champ moyen. Cette quantité est appelée le Tenseur de Reynolds.

LES : L'approche dite *filtrée*, consiste à décomposer toutes les variables apparaissant dans les équations de Navier-Stokes en une somme d'une composante filtrée et d'une composante résiduelle. De façon analogue aux modèles RANS, ces procédures résolvent les équations de la MDF filtrées et modélisent les interactions entre les composantes filtrées, dites résolues, et résiduelles, non résolues, à travers le tenseur de contraintes sous mailles

Dans ces deux approches, la modélisation des injections de quantité de mouvement par les

4. Landau and Lifshitz (1987)

termes fluctuants, ou résiduels est nécessaire. Cette modélisation “ferme” le système composé des équations simplifiées et de l’équation de la conservation de la masse issue de la mécanique des milieux continus (voir plus bas). La fermeture de ce système consiste tout simplement à fournir une relation supplémentaire liant les quantités inconnues, propres à la modélisation choisie, avec les quantités disponibles dans ce cadre. Sans cette fermeture, il est impossible de résoudre le système. Le challenge est d’établir cette équation supplémentaire ; on parle de problème de fermeture.

Alors que ces modèles se développent et se complexifient, une multitude de données considérées comme exactes, puisque obtenues sans modèle, ont été générées par les simulations directes et les expériences menées pendant des décennies. En parallèle, les algorithmes d’assimilation de données puis d’intelligence artificielle ont permis l’essor de nouveaux domaines de recherche, et de rentabiliser une nouvelle fois les travaux déjà effectués. Les méthodes d’assimilation de données consistent essentiellement en un problème d’optimisation. Le but est d’ajuster un ensemble de paramètres \mathbf{W} pour pouvoir modéliser une ou plusieurs sortie.s \mathbf{Y} à partir d’un corpus de données \mathbf{X} fourni à l’algorithme. La fiabilité de cette modélisation est évaluée selon *une fonction de coût* et parfois une ou plusieurs *métrique.s* supplémentaire.s permettant d’avoir une appréhension plus globale de la qualité de la sortie fournie. L’intelligence artificielle (IA) repose sur le même principe sauf qu’il est nécessaire que cette optimisation soit assez générale pour pouvoir être appliquée à des cas inédits sans en modifier les paramètres préalablement établis. Dans ce contexte, l’optimisation de ces paramètres est appelée *apprentissage* ou *entraînement*. L’apprentissage, indispensable, doit permettre à l’IA de *prédire* avec une efficacité proche de celle évaluée en fin d’entraînement.

Il existe plusieurs formes d’intelligence artificielle. La plus connue, la plus célèbre et la plus puissante à ce jour est le réseau de neurones. Il en existe plusieurs types : les réseaux de neurones classiques qui consistent en une succession de transformations non-linéaires des entrées effectuées par des unités de calculs appelées *neurones* dans le but de prédire les sorties attendues. Les sorties de chaque neurone sont pondérées par les paramètres appelés dans ce contexte *poinds*. Ils sont optimisés au cours de l’apprentissage pour minimiser les erreurs commises lors de la prédiction. Il existe des réseaux de neurones à convolutions qui contrairement à la catégorie précédente s’applique à des données présentant une dimension spatiale (un vecteur, une image, un volume). Ces réseaux ne cessent de se surpasser et leurs applications sont de plus en plus variées. Au vue des excellents résultats de ces méthodes, de leur popularité et de leur accessibilité croissantes, beaucoup de chercheurs ont travaillé sur le couplage des méthodes d’assimilation de données et de l’IA avec l’étude de la Turbulence, en particulier pour élaborer des modèles RANS ou LES *augmentés* par les données. L’idée générale est la suivante : les données issues de simulations directes ou d’expériences renferment toute l’information nécessaire pour améliorer les modèles existants, intrinsèquement déficients. Les chercheurs substituent les hypothèses émises lors de l’établissement des modèles, par une injection d’informations issues de données exactes. Certains travaux essaient d’ajuster les paramètres des modèles pré-existants, d’autres les champs modélisés eux mêmes, et plus rarement, de prédire le champ en question sans le modéliser. Dans la majorité des cas, les travaux s’affranchissent de la cohérence spatiale des structures au sein des champs turbulents et propose d’effectuer des prédictions point par point.

D’un point de vue fondamentale et conceptuelle, l’étude de la turbulence dans les écoulements

du monde qui nous entoure est souvent bien trop complexe ; il est alors nécessaire de considérer des cas idéaux pour en appréhender et tenter de comprendre les phénomènes en jeu, en dégager des grandeurs adimensionnelles clés (comme le Reynolds) etc ⁵. Nous considérerons une turbulence homogène : les moments statistiques (moyennes, variances etc.) des fonctions des vitesses, ou d'autres grandeurs turbulentes, ainsi que de leurs dérivées sont invariantes par toute translation arbitraire. De plus, nous supposons également que la turbulence est isotrope, c'est à dire que les grandeurs citées précédemment sont supposées invariantes par rotations. Nous nous plaçons donc dans le cas d'une turbulence homogène isotrope, ou HIT (en anglais). Les connaissances acquises en HIT constituent un socle de réflexion ouvrant la voie à l'étude de champs plus complexes, moins idéaux. Le fluide est supposé incompressible c'est-à-dire que la déformation du fluide considéré se fait sans altérer son volume. De même, nous considérons un domaine cubiques de dimensions $2\pi \times 2\pi \times 2\pi$ aux conditions aux limites tri-périodiques permettant l'utilisation de l'analyse de Fourier et les méthodes de résolutions numériques associées.

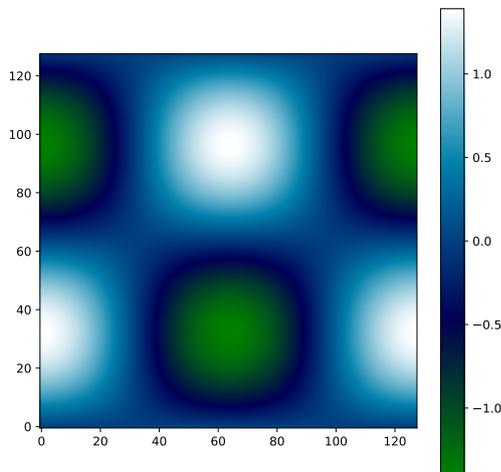
Les travaux que nous proposons dans ce manuscrit se situent au carrefour de la mécanique des fluides et de l'intelligence artificielle. Dans le cadre d'une simulation LES, nous voulons prédire le tenseur des contraintes sous mailles à partir du champ de vitesse filtré. Exprimé d'une autre façon, nous voulons fournir une fermeture aux équations de la LES sans aucune hypothèse ou loi qui contraindraient l'étendue de cette relation. Nous entendons utiliser la force pure de l'intelligence artificielle en effectuant des prédictions de grandeurs turbulentes inconnues, en volume, sans schémas numériques élaborés. Le champ de vitesse filtré en entrée ne sera pas substantiellement modifié et les six composantes différentes du tenseur des contraintes seront prédites simultanément par ce même réseau. La motivation essentielle de ces travaux est de montrer qu'à travers les données, l'IA peut permettre de fermer les systèmes d'équations des approches considérées, tout en tenant compte d'une physique plus riche et complète que ne le permettent les modèles existants. Dans un second temps, nous voulons tester si la relation apprise au préalable, peut se généraliser à des écoulements intrinsèquement différents de ceux considérés pendant l'entraînement. En ce sens, nous voulons répondre aux questions fondamentales relatives à la mécanique des fluides et à l'IA : est ce qu'il existe une relation entre le tenseur des contraintes sous mailles et le champ de vitesse, est ce qu'un réseau de neurones peut apprendre et généraliser sur des problèmes aussi complexes que ceux que nous traiterons.

Ce manuscrit se compose de quatre chapitres en plus de l'introduction et de la conclusion. Le corps de cet ouvrage s'ouvre sur une étude des mathématiques et des concepts liés à l'étude de la mécanique des fluides et de la Turbulence plus particulièrement, chapitre 2. Nous y aborderons de manière plus formelle le principe de la cascade énergétique en utilisant l'analyse de Fourier préalablement introduite. Nous y verrons concrètement le phénomène de l'advection et en quoi cette non linéarité renferme toute la complexité de la dynamique des fluides. Dans une troisième section, l'approche LES, le tenseur des contraintes sous-mailles et deux des modèles les plus utilisées seront détaillés. Enfin, fort de ce formalisme et de ces modèles, nous introduirons les outils statistiques avec lesquels les modélisations/prédictions seront évaluées ainsi que les grandeurs calculées presque systématiquement dans le reste du document. Le chapitre suivant posera les bases de tous les concepts sous-jacents à l'élaboration des réseaux de neurones et en particulier des réseaux de neurones à convolutions. En oscillant entre description formelle et

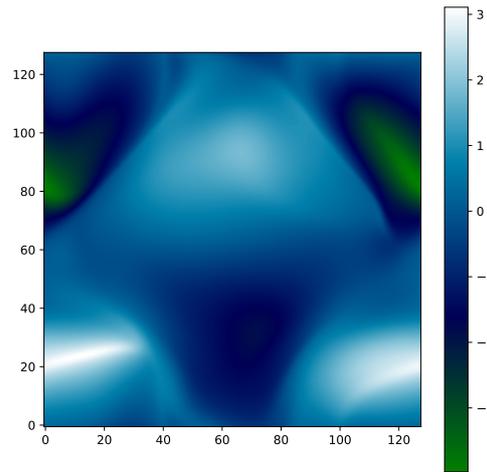
5. Gence (1987).

interprétation, nous introduirons des concepts comme l'apprentissage supervisée, l'estimation ponctuelle, la pénalisation de la fonction de coût etc. et ce qui en découle. Le chapitre 4 est le chapitre charnière entre les deux précédents et le suivant. En effet, dans un premier temps, nous établirons une revue non exhaustive des différents types de méthodes liant intelligence artificielle et turbulence de la littérature, ainsi que leurs apports principaux. Puis le reste du chapitre sera consacré à notre méthode, les raisons qui motivent ce choix et à sa structure. Nous illustrerons l'optimisation des hyper-paramètres inhérents aux méthodes d'IA et en quoi il est capitale d'élargir au maximum cette recherche. Enfin, après avoir entraîné notre modèle sur une base de données détaillée et construite de façon intuitive et justifiée physiquement, nous évaluerons l'apprentissage de ce dernier tant sur l'ensemble de test que sur une itération complète. Le cinquième et dernier chapitre avant la conclusion, nous testons les capacités de généralisation du modèle présenté au chapitre précédent. En utilisant les méthodes d'évaluation proposées chapitre 2, nous montrons en quoi le modèle construit surpasse à différent niveaux les modèles algébriques introduits, et nous essayerons d'établir à quel point l'IA permet de découvrir la Physique sous-jacente reliant le champ de vitesse de le tenseur des contraintes.

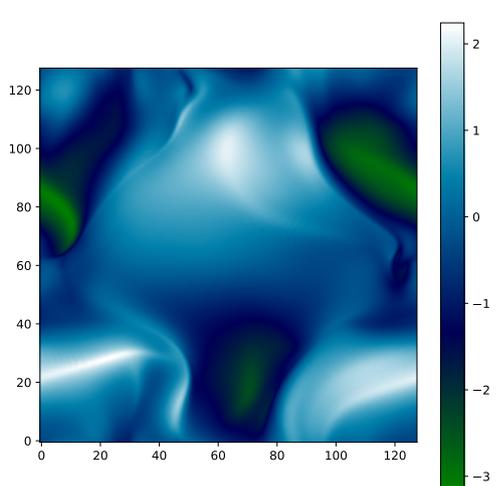
Nous espérons que la lecture de ce manuscrit sera claire et agréable.



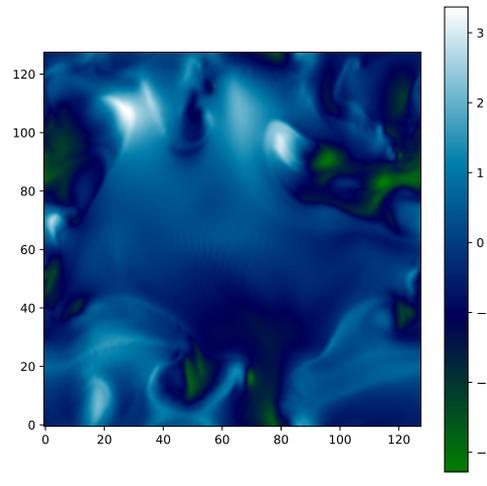
(a) Initialisation du champ de vitesse de type Taylor-Green



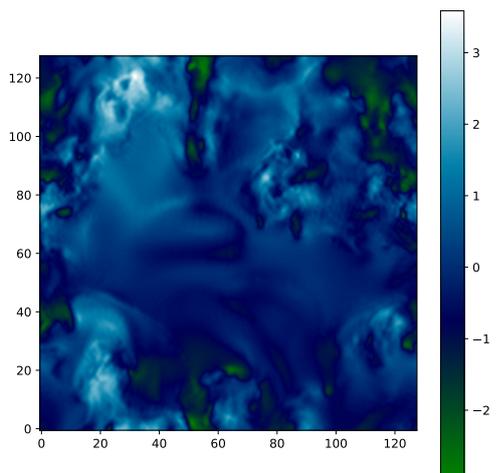
(b) Évolution de l'écoulement apparition des instabilités



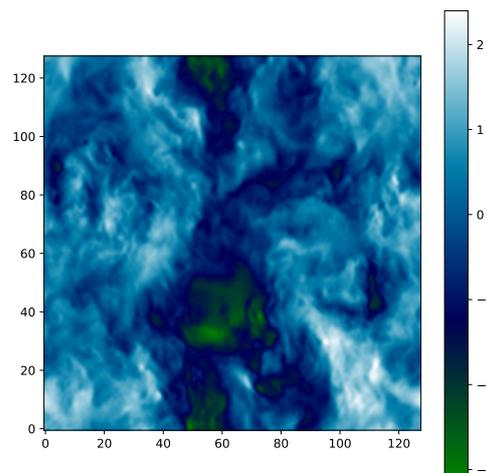
(c) Les grandes échelles commencent à se décomposer.



(d) Plus loin en temps, les instabilités se développent, les patterns turbulents apparaissent.



(e) Les petites structures se diffusent dans l'ensemble du domaine. Les structures initiales disparaissent



(f) La turbulence est diffuse et développée, les échelles sont multiples intriquées et non distinguables

Figure 1.5 – Évolution temporelle de la Turbulence au sein d'un écoulement.

Chapitre 2

Méthodes LES pour la Turbulence ; grandeurs et outils pour l'analyse des modèles

L'essence de la mécanique des fluides réside dans ses équations décrivant la totalité des écoulements et de leur devenir au cours du temps. Ce chapitre pose les bases non seulement de la mécanique des fluides et de la turbulence mais également des approches et des outils que nous considérerons avec plus de détails dans les chapitres suivants. Il se décompose en quatre parties de tailles disparates : au cours d'une première partie, nous écrirons les équations de Navier-Stokes dans l'espace physique en rappelant les hypothèses générales pour leurs obtentions. Dans une seconde partie, nous nous intéresserons à l'interprétation des équations de Navier-Stokes pour la description de la turbulence : de sa phénoménologie aux hypothèses de Kolmogorov, mettant en exergue la cascade d'énergie décrivant le processus de transfert d'énergie des grandes aux petites échelles, jusqu'à dissipation. Nous introduirons les outils statistiques comme les corrélations et introduirons l'analyse de Fourier pour mener à bien les objectifs de cette section. Au cours d'une troisième partie, nous introduirons les méthodes de simulations des grandes échelles et notamment les modèles de Smagorinsky et de Clark ainsi que leurs limites à partir de la littérature. Nous introduirons des grandeurs qui seront le centre de toute l'étude au cœur de nos travaux, au chapitre 5. Enfin dans une quatrième et dernière section nous présenterons des outils d'analyse et de mesure de performances des modèles sus-mentionnés ainsi que des modèles du chapitre précédemment cité.

2.1 Équations constitutives de la mécanique des fluides

Dans cette section, nous considérons que les grandeurs varient en fonction du temps noté t et de l'espace \boldsymbol{x} . Nous ne mentionnerons pas ces variables pour ne pas alourdir la lecture.

Conservation de la masse - L'hypothèse de milieu continu nous permet de définir des grandeurs (continues) propres au fluide comme sa masse volumique ρ , sa vitesse dans l'espace U_i avec $i = 1, 2, 3$, le champ de pression p etc en tout point de l'espace. Pour comprendre

l'écoulement, il est nécessaire de le modéliser théoriquement par des équations qui le régissent. La mécanique des milieux continus nous permet d'écrire une première équation constitutive de la mécanique des fluides, la conservation de la masse¹ :

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho U_i)}{\partial x_i} = 0 \quad (2.1.1)$$

Nous nous plaçons dans l'étude des fluides incompressibles c'est-à-dire que le volume de la particule fluide est invariant au cours du temps et de son évolution dans l'écoulement, cette invariance est à comprendre au sens de Lagrange, nous écrivons donc

$$\frac{DV}{Dt} = 0 \Leftrightarrow \frac{D(m/\rho)}{Dt} = 0 \quad (2.1.2)$$

$$\Leftrightarrow -\frac{m}{\rho^2} \frac{D\rho}{Dt} = 0 \quad (2.1.3)$$

$$\Leftrightarrow \frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + U_i \frac{\partial \rho}{\partial x_i} = 0 \quad (2.1.4)$$

Si nous injectons le résultat obtenu Eq.(2.1.4) dans Eq.(2.1.1), nous obtenons l'équation de continuité

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (2.1.5)$$

L'équation Eq.(2.1.4) n'implique pas *ipso facto* que la masse volumique d'un fluide incompressible est forcément constante, néanmoins nous **nous plaçons dans le cas particulier où la masse volumique est constante.**

Conservation de la quantité de mouvement - Une autre implication des milieux continus est la suivante : le taux de variation de la quantité de mouvement d'une particule fluide d'un temps t au temps $t + dt$ dépend uniquement des forces qui s'appliquent sur cette particule fluide ; c'est le second principe fondamentale de la dynamique. Ainsi en effectuant un bilan de force sur une particule fluide, nous pouvons écrire trois relations constitutives de la mécanique des fluides. Nous considérons deux types de forces : des forces volumiques F qui s'appliquent en tout point du volume ; et les forces surfaciques qui s'appliquent aux frontières de ce volume. Ces dernières forces sont agglomérées dans le tenseur σ_{ij} . Il est possible alors d'écrire la seconde loi de Newton sur notre particule fluide au cours de son mouvement dans l'écoulement

$$\rho \frac{DU_i}{Dt} = \rho F_i + \frac{\partial \sigma_{ij}}{\partial x_j} \quad (2.1.6)$$

Le tenseur des contraintes - C'est Cauchy en 1822 qui introduit le concept de tenseur des contraintes. À chaque instant, les molécules contenues dans une particule fluide vont traverser ses surfaces et générer une force au niveau de ces surfaces. La forme adéquate pour modéliser ces contraintes est un tenseur d'ordre 2 puisque toutes ces contraintes peuvent se projeter dans le repère de l'écoulement. En effet, chaque surface par laquelle le flux de molécules traverse se repère par une normale (sortante par rapport à la particule fluide) qui elle s'exprime par

1. La répétition d'indice indique une sommation, conformément à la notation d'Einstein

au maximum deux des vecteurs unitaires de la base canonique du problème. Les contraintes exprimées par un seul vecteur unitaire sont les composantes diagonales du tenseur σ_{ij} et les six autres en constituent les composantes extradiagonales. Les composantes diagonales sont appelées les composantes isotropes et résultent du champ de pression au sein de l'écoulement². Les composantes extradiagonales composent le tenseur à diagonale nulle d_{ij} appelés le tenseur déviatorique. Elles sont le résultat de la non uniformité du champ de vitesse dans le continuum et en particulier dans le voisinage de la particule fluide étudiée (Batchelor, 2000). En effet, Batchelor explique que ces flux de quantités de mouvement (décrits par d_{ij}) entraîne des interactions de friction au sein de la matière transportée, c'est ce que l'on appelle le cisaillement. Puisque ce sont les variations du champs de vitesse local qui sont à l'origine de ces flux, l'approche phénoménologique suggère que le terme le plus impactant dans la générations des contraintes d_{ij} est le gradient de vitesse $\partial u_i / \partial x_j$. Enfin Batchelor (2000) p.143 établit une relation tensorielle entre d_{ij} et le gradient de vitesse en affirmant que "nous n'avons aucune façon de déduire la dépendance de d_{ij} en fonction de $\partial u_i / \partial x_j$ pour les fluides en général" ; la relation se base sur des hypothèses phénoménologique comme mentionné plus haut et établit la formule suivante

$$d_{ij} = A_{ijkl} \frac{\partial U_k}{\partial x_l} \quad (2.1.7)$$

où A_{ijkl} un tenseur de coefficients d'ordre 4 qui renferme l'état local du fluide, tenseur symétrique de même que d_{ij} l'est. La linéarité en $\partial U_i / \partial x_j$ de cette équation se base également sur une hypothèse qui suppose que localement "les composantes du gradient sont de magnitude suffisamment faible".

Enfin, en écrivant le gradient de vitesse comme la somme du tenseur de déformation S_{ij} et du tenseur de rotation solide Ω_{ij} ³, en supposant le fluide homogène et isotrope, Poisson en 1831 et Stokes en 1845 obtiennent la forme de d_{ij} suivante (dans le cas incompressible) :

$$d_{ij} = 2\mu S_{ij} \quad \text{avec} \quad S_{ij} = \frac{1}{2} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \quad (2.1.8)$$

μ est nécessairement positif et constitue une propriété du fluide étudié. Il caractérise la capacité du fluide à aplanir les variations spatiales du champ de vitesse moyen. Il est appelé viscosité dynamique en ce sens qu'il s'oppose aux déformations induites par le gradient de vitesse local, prenant donc en compte les flux (on le voit dans sa dimension $M L^{-1} T^{-1}$ qui implique la masse). Le tenseur de contraintes σ_{ij} s'écrit finalement

$$\sigma_{ij} = -P\delta_{ij} + 2\mu S_{ij} \quad (2.1.9)$$

Enfin nous supposons que les variations de viscosité ne sont pas significatives pour les prendre en compte, on se place dans le fluide Newtonien. Alors, en combinant les équations (2.1.5), (2.1.6) et (2.1.9), nous obtenons les équations de Navier-Stokes pour un fluide Newtonien incompressible :

2. Il suffit alors de considérer un fluide au repos pour faire apparaître l'équation de l'hydrostatique à partir de Eq.(2.1.6)

3. Une note p.144 stipule qu'une intuition pousse à choisir les composante de A_{ijkl} telles qu'elles annulent la participation des composantes de Ω_{ij} dans le processus générant d_{ij}

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 U_i}{\partial x_j \partial x_j} + F_i \quad (2.1.10a)$$

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (2.1.10b)$$

La quantité $\nu = \mu/\rho$ est la viscosité cinématique, il traduit la diffusivité de la quantité de mouvement du fluide. Au plus cette quantité est élevée au plus les non-uniformités du champ de vitesse seront “absorbées” par les particules composant le fluide. Une analyse de la viscosité peut également se faire à partir du bilan d’énergie interne de la particule fluide voir l’analyse de [Batchelor \(2000\)](#) à ce sujet.

Dans l’équation précédente, le terme de gauche $U_j \partial U_i / \partial x_j$ est non linéaire. Il traduit l’injection d’énergie cinétique par le fluide entourant la particule fluide, dans cette dernière. Ce terme est le terme le plus riche de l’équation de Navier-Stokes et en même temps le plus compliqué à modéliser numériquement à cause la non linéarité. Le terme faisant apparaître la dérivée seconde traduit les effets visqueux qui vont avoir tendance à dissiper la quantité de mouvement. Le rapport de ces deux termes constituent le nombre de Reynolds comme nous le verrons dans la section suivante [2.2](#).

Équation de l’énergie cinétique - Cette équation peut être obtenue en multipliant l’équation Eq.([2.1.10a](#)) par la quantité U_i . Dans un premier temps on écrit U_i ([2.1.10a](#)) :

$$U_i \frac{\partial U_i}{\partial t} + U_i U_j \frac{\partial U_i}{\partial x_j} = -\frac{U_i}{\rho} \frac{\partial p}{\partial x_i} + \nu U_i \frac{\partial^2 U_i}{\partial x_j \partial x_j} + U_i F_i \quad (2.1.11)$$

Puis en remarquant que

$$U_i \frac{\partial U_i}{\partial t} = \frac{\partial U_i^2 / 2}{\partial t} \quad \text{et} \quad U_i U_j \frac{\partial U_i}{\partial x_j} = U_j \frac{\partial U_i^2 / 2}{\partial x_j}$$

le terme de gauche devient alors

$$\frac{DU_i^2 / 2}{Dt} = \frac{DE}{Dt} \quad (2.1.12)$$

Avec $E = 1/2(U_1^2 + U_2^2 + U_3^2)$ l’énergie cinétique transportée par la particule fluide.

Le terme visqueux à droite peut lui aussi être réécrit puisque :

$$\nu \frac{\partial}{\partial x_j} \left(U_i \frac{\partial U_i}{\partial x_j} \right) = \nu \left[\frac{\partial U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} + U_i \frac{\partial^2 U_i}{\partial x_j^2} \right] \quad (2.1.13)$$

Et donc

$$U_i \nu \frac{\partial^2 U_i}{\partial x_j^2} = \nu \left[\frac{\partial}{\partial x_j} \left(U_i \frac{\partial U_i}{\partial x_j} \right) - \frac{\partial U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} \right] \quad (2.1.14)$$

$$= \nu \left[\frac{\partial^2 U_i^2 / 2}{\partial x_j^2} - \frac{\partial U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} \right] \quad (2.1.15)$$

L'équation de l'évolution de l'énergie cinétique s'écrit enfin

$$\boxed{\frac{DE}{Dt} = -\frac{\partial U_i P}{\partial x_i} + \nu \frac{\partial^2 E}{\partial x_j \partial x_j} - \nu \frac{\partial U_i}{\partial x_j} \frac{\partial U_i}{\partial x_j} + U_i F_i} \quad (2.1.16)$$

Enfin, un mot sur le champ de pression P . Or par l'incompressibilité du fluide, on peut montrer (Pope, 2001) que ce champ vérifie l'équation de Poisson :

$$\nabla^2(P/\rho) = -\frac{\partial U_i}{\partial x_j} \frac{\partial U_j}{\partial x_i} \quad (2.1.17)$$

En utilisant le formalisme des fonctions de Green, on peut alors démontrer que la pression est une réaction du fluide incompressible pour assurer que l'équation Eq.(2.1.10a) vérifie la condition d'incompressibilité, c'est donc un multiplicateur de Lagrange.

Nous clôturons cette section en soulignant que les équations (2.1.10a), (2.1.10b) et (2.1.16) forment les équations générales qui régissent la mécanique des fluides, et notamment les écoulement appartenant au régime régime turbulent.

2.2 Phénoménologie de la turbulence

Définir les équations et rappeler les phénomènes qu'elles expriment étaient des étapes indispensables, bien que théoriques et générales. Dans cette section, nous décrivons avec plus de précision et de profondeur la phénoménologie de la turbulence. Cette compréhension est fondamentale pour comprendre la philosophie des modèles de simulations aux grandes échelles.

2.2.1 Le nombre de Reynolds et hypothèses de Kolmogorov

Rappelons les équations de Navier-Stokes :

$$\frac{\partial U_i}{\partial t} + \underbrace{U_j \frac{\partial U_i}{\partial x_j}}_{\text{Inertie - terme non linéaire}} = -\frac{1}{\rho} \frac{\partial P}{\partial x_j} + \underbrace{\nu \frac{\partial^2 U_i}{\partial x_j^2}}_{\text{Visqueux}} \quad (2.2.1)$$

$$\frac{\partial U_i}{\partial x_i} = 0 \quad (2.2.2)$$

Osborne Reynolds construisit en 1883 un nombre sans dimension comparant les effets d'inertie sur les effets visqueux; il est appelé nombre de Reynolds, abrégé Reynolds; il peut s'écrire comme

$$\text{Re} = \left(U_j \frac{\partial U_i}{\partial x_j} \right) / \left(\nu \frac{\partial^2 U_i}{\partial x_j^2} \right) = \frac{\text{Effets inertiels}}{\text{Effets visqueux}} = \frac{\mathcal{U}\mathcal{L}}{\nu}$$

avec \mathcal{U} and \mathcal{L} les échelles de vitesses et de longueur propres à la structure tourbillonnaire. Nous l'avons dit, plus le Reynolds est élevé et plus grande sera la variété de structures tour-

billonnaires composant l'écoulement. À cause des variations du champ de vitesse locale, ces structures interagissent entre elles se déformant par étirement/contraction. D'après Richardson et Kolmogorov, ce sont ces interactions qui donnent naissance à des échelles plus petites en tailles et moins énergétiques. À partir de la longueur caractéristique de la structure et de la viscosité de l'écoulement, nous pouvons construire son temps caractéristique de déformation :

$$\tau_{\text{def}} \sim \frac{\mathcal{L}}{\sqrt{U^2}}$$

Nous pouvons également estimer le temps de dissipation de l'énergie τ_{diss} par frottements en fonction de la taille et de la vitesse de la structure

$$\tau_{\text{diss}} \sim \frac{\mathcal{L}^2}{\nu}$$

Il est intéressant de constater que leur rapport fait apparaître le nombre de Reynolds associé à cette échelle :

$$\frac{\tau_{\text{diss}}}{\tau_{\text{def}}} \sim \frac{U\mathcal{L}}{\nu} = \text{Re}_{\mathcal{L}} \Leftrightarrow \tau_{\text{diss}} \sim \tau_{\text{def}} \text{Re}_{\mathcal{L}}$$

Il apparaît alors qu'à partir du moment où le Reynolds associé à l'échelle étudiée est élevé, le processus de déformation/contraction et donc création de structures plus petites s'effectuera beaucoup plus vite que la dissipation moléculaire. L'énergie cinétique turbulente est donc quasiment intégralement conservée dans ce processus de cascade, jusqu'à ce que les deux temps deviennent de même ordre de grandeur. Cette égalité est atteinte pour les structures de très petites tailles, donc peu énergétiques ; les déformations nécessitent autant voire plus de temps, et l'énergie véhiculée par ces structures est dissipée par frottements. Kolmogorov conclut que le taux de dissipation de l'énergie cinétique dans ces petites échelles est sensiblement le même que l'injection d'énergie dans le système à travers les plus grandes structures tourbillonnaires.

Kolmogorov (Kolmogorov, 1941) émit d'autres hypothèses qui sont encore au cœur de la théorie de la turbulence, il convient d'en rappeler l'essentiel. L'une de ces hypothèses est l'isotropie locale. Elle exprime que dès lors que le Reynolds de l'écoulement est assez élevé, la dynamique des petites échelles obéira de manière universelle au formalisme de la turbulence homogène isotrope. Elle implique également que les anisotropies (s'il y en a dans l'écoulement), n'agissent que sur la dynamique des grandes échelles. Au fur et à mesure que l'énergie cinétique turbulente se distribue au sein des petites échelles, le fluide amortit les anisotropies (en les répartissant) jusqu'à atteindre le régime isotrope des petites échelles. Aux plus petites de ces échelles, la dynamique est totalement pilotée par deux grandeurs : la viscosité ν et le taux de dissipation de l'écoulement ε . Il construit les plus petites échelles pouvant être observé dans un écoulement donné à partir de raisonnements sur les ordres de grandeurs, il définit les échelles dites de Kolmogorov :

$$\eta = \left(\frac{\nu^3}{\langle \varepsilon \rangle} \right)^{1/4}, \quad t_\eta = \left(\frac{\nu}{\langle \varepsilon \rangle} \right)^{1/2}, \quad v_\eta = (\nu \langle \varepsilon \rangle)^{1/4} \quad (2.2.3)$$

avec η , t_η et v_η la taille, le temps et la vitesse caractéristiques associés à ces échelles. On remarque également que le Reynolds associé est $\text{Re}_\eta = 1$. À partir d'ici, la dissipation moléculaire est plus rapide que les mécanismes d'étirement/contraction et l'énergie cinétique véhiculée est totalement transformée en chaleur par frottements moléculaires. C'est pour cette raison que

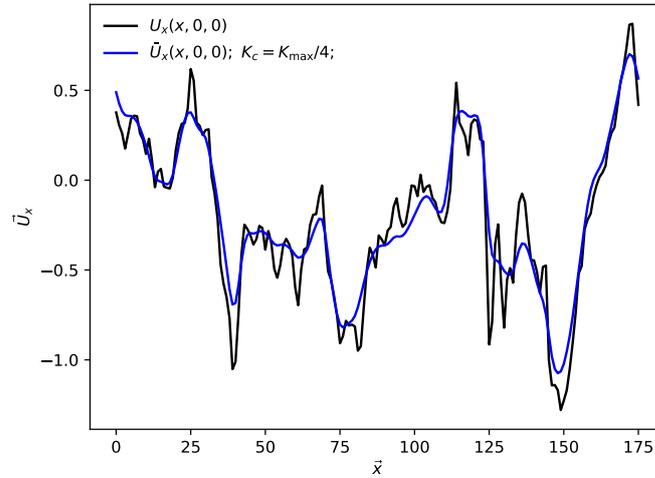


Figure 2.1 – Deux dynamiques au sein du champ de vitesse. La courbe noire représente le champ de vitesse $U_x(x, 0, 0)$; la courbe bleue représente cette même grandeur dont les hautes fréquences ont été coupées.

les échelles de Kolmogorov sont les plus petites permises par la turbulence. Une dernière hypothèse que nous souhaitons rappeler pour la compréhension de la cascade au sens de Kolmogorov, est un résultat que nous avons “démonstré” précédemment : lorsque le Reynolds est suffisamment grand, il existe une gamme d’échelle intermédiaire entre les grandes échelles et les échelles dissipatives, pour lesquelles la dynamique est pilotée uniquement par le taux d’injection d’énergie/dissipation ε et indépendant des effets visqueux pilotés par ν . Cette gamme appelée gammes inertielles ou gamme énergétique (Pope, 2001).

Notons enfin, que le lien entre taux d’injection d’énergie cinétique aux grandes échelles et taux de dissipation par les petites échelles implique la possibilité de définir les échelles de Kolmogorov à partir des grandes échelles :

$$\eta = \frac{L}{\text{Re}_L^{3/4}}, \quad t_\eta = \frac{T}{\text{Re}_L^{1/2}}, \quad v_\eta = \frac{V}{\text{Re}_L^{1/4}} \quad (2.2.4)$$

échelles, et L , T et V celles des grandes échelles. Le terme Re_L représente ici le nombre de Reynolds associé aux plus grandes structures tourbillonnaires de l’écoulement. Nous pouvons alors dire, que ces hypothèses s’appliquent dès lors que Re_L est grand de telle sorte que

$$\eta \ll L \quad t_\eta \ll T \quad v_\eta < V$$

2.2.2 Corrélation et analyse spectrale de l’écoulement

L’incapacité de l’écoulement à dissiper les fluctuations du champ de vitesse en son sein rend l’écoulement chaotique et non-prédictible. Mesuré, le champ de vitesse ressemble alors un signal bruité dans lequel il est possible de dégager deux comportements, deux dynamiques : une

évolution globale (grande longueur d'onde) sur laquelle se superposent des oscillations plus rapides, voir Fig.(2.1). La figure (1.4) illustre la topologie complexe du champ de vorticit   l'autre face de la vitesse⁴. Ces diff  rentes remarques poussent    consid  rer l'  coulement avec des outils statistiques comme la corr  lation moyenne en deux points, et particuli  rement la corr  lation du champ de vitesse. Le tenseur de corr  lations des vitesses Γ aussi appel   covariance, s'  crit comme la moyenne du produit des vitesses    deux points s  par  s d'un vecteur $\mathbf{r} \in \mathbb{R}^3$:

$$\Gamma_{ij}(\mathbf{x}, \mathbf{r}, t) = \langle U_i(\mathbf{x}, t) U_j(\mathbf{x} + \mathbf{r}, t) \rangle \quad (2.2.5)$$

Notons qu'en posant $\mathbf{r} = \mathbf{0}$, et en consid  rant $i = j$ c'est-  -dire l'auto-corr  lation du champ de vitesse, on retrouve l'  nergie cin  tique $\Gamma_{ii}(\mathbf{x}, 0, t) = 2E(\mathbf{x}, t)$.

  tudier la dynamique de cette corr  lation peut   tre fait (De Karman and Howarth (1938), Hinze (1975) et Pope (2001) entre beaucoup d'autres), mais cela reste insuffisant pour en d  gager une ph  nom  nologie claire. Pour atteindre cet objectif il faut examiner l'  quation de Navier-Stokes dans l'espace de Fourier et des nombre d'ondes. Dans la repr  sentation de Fourier, nous pouvons d  composer le champ de vitesse comme la somme des modes le constituant, ils sont not  s $e^{i\kappa_j x_j}$ avec κ_j les composantes du vecteur d'onde $\boldsymbol{\kappa}$ et $j = 1, 2, 3$. On peut   crire le champ de vitesse comme une s  rie de Fourier :

$$U(\mathbf{x}, t) = \sum_{\boldsymbol{\kappa}} e^{i\kappa_j x_j} \hat{u}(\boldsymbol{\kappa}, t) \quad (2.2.6)$$

la sommation se faisant sur un nombre infini de nombre d'ondes discrets. Les coefficients de Fourier $\hat{u}_j(\boldsymbol{\kappa}, t)$ sont d  finis par la transform  e de Fourier (supposant le domaine de longueur 2π , tri-p  riodique) (Pope, 2001)

$$\hat{u}_j(\boldsymbol{\kappa}, t) = \left(\frac{1}{2\pi}\right)^3 \int_0^{2\pi} \int_0^{2\pi} \int_0^{2\pi} e^{-i\kappa_l x_l} U_j(\mathbf{x}, t) dx_1 dx_2 dx_3 \quad (2.2.7)$$

Chaque coefficient $\hat{u}(\boldsymbol{\kappa}, t)$ est d  pendant du temps et est   galement al  atoire. Les modes sont quant    eux constants en fonction du temps ; aussi les fluctuations au cours du temps des coefficients portent toute l'information du champ de vitesse. On peut alors   crire les   quations de Navier-Stokes pour chaque coefficient en transformant terme    terme l'  quation Eq.(2.1.10a) en utilisant la lin  arit   de l'op  rateur int  gral et la propri  t   permettant d'  crire la transform  e de Fourier d'un produit de deux grandeurs temporelles, comme la convolution des deux transform  es respectives des deux termes. Ainsi le terme non lin  aire $U_j \partial U_k / \partial x_j = \partial(U_k U_j) / \partial x_j$ peut s'  crire :

$$\mathcal{F} \left\{ \frac{\partial U_k U_j}{\partial x_j} \right\} = i\kappa_j \sum_{\boldsymbol{\kappa}'} \sum_{\boldsymbol{\kappa}''} \hat{u}_k(\boldsymbol{\kappa}') \hat{u}_j(\boldsymbol{\kappa}'') \quad (2.2.8)$$

$$= i\kappa_j \sum_{\boldsymbol{\kappa}'} \hat{u}_k(\boldsymbol{\kappa}', t) \hat{u}_j(\boldsymbol{\kappa} - \boldsymbol{\kappa}', t) \quad (2.2.9)$$

Avec $\boldsymbol{\kappa}''$ choisi tel que $\boldsymbol{\kappa} = \boldsymbol{\kappa}' + \boldsymbol{\kappa}''$. L'indice k dans l'  quation pr  c  dent est muet et est r  p  t  . Une attention particuli  re doit   tre   galement port  e sur le terme de pression. En effet, nous

4. Le champ de vorticit   est    la vitesse ce que le champ magn  tique est pour le champ   lectrique. Ils sont perpendiculaires en tout points et    tout instant.

avons établi Eq.(2.1.17) pouvant se réécrire

$$-\nabla^2(P/\rho) = \frac{\partial}{\partial x_k} \left(\frac{\partial}{\partial x_j} (U_k U_j) \right) \quad (2.2.10)$$

En décomposant P/ρ , la pression dynamique, en série de Fourier de manière analogue à Eq.(2.2.6), on définit $\hat{p}(\boldsymbol{\kappa}, t)$ les coefficients de Fourier de P/ρ , et

$$\hat{p}(\boldsymbol{\kappa}, t) = -\frac{\kappa_j \kappa_k}{\kappa^2} \sum_{\boldsymbol{\kappa}'} \hat{u}_j(\boldsymbol{\kappa}', t) \hat{u}_k(\boldsymbol{\kappa} - \boldsymbol{\kappa}', t) \quad (2.2.11)$$

$$= i\kappa_k \mathcal{F} \left\{ \frac{\partial U_k U_j}{\partial x_j} \right\} \quad (2.2.12)$$

avec $\kappa^2 = \kappa_j \kappa_j$. Puisque le terme non-linéaire apparaît également dans l'équation de la pression, nous définissons le tenseur commun appelé projecteur

$$P_{jk} = \left(\delta_{jk} - \frac{\kappa_j \kappa_k}{\kappa^2} \right) \quad (2.2.13)$$

Les transformées de Fourier du terme visqueux Eq.(2.1.10a) ainsi que de l'équation de continuité Eq.(2.1.10b) se calculent directement et

$$\mathcal{F} \left\{ \nu \frac{\partial^2 U_j}{\partial x_k \partial x_k} \right\} = -\nu \kappa^2 \hat{u}_j \quad \text{et} \quad \mathcal{F} \left\{ \frac{\partial U_j}{\partial x_j} \right\} = -i\kappa_j \hat{u}_j \quad (2.2.14)$$

Finalement, nous obtenons les équations de Navier-Stokes pour le coefficient associé au mode de vecteur d'onde $\boldsymbol{\kappa}$ fixé :

$$\left(\frac{\partial}{\partial t} + \nu \kappa^2 \right) \hat{u}_j(\boldsymbol{\kappa}, t) = -i\kappa_l P_{jk}(\boldsymbol{\kappa}) \sum_{\boldsymbol{\kappa}'} \hat{u}_k(\boldsymbol{\kappa}', t) \hat{u}_l(\boldsymbol{\kappa} - \boldsymbol{\kappa}', t) \quad (2.2.15a)$$

$$\kappa_j \hat{u}_j(\boldsymbol{\kappa}, t) = 0 \quad (2.2.15b)$$

Si à gauche du signe égal de l'équation Eq.(2.2.15a), le terme ne fait intervenir qu'un seul nombre d'onde $\boldsymbol{\kappa}$, le terme de droite fait intervenir les modes de \hat{u} à trois nombres d'onde différents, vérifiant la condition $\boldsymbol{\kappa} = \boldsymbol{\kappa}' + \boldsymbol{\kappa}''$. Ce terme représente la convection dans l'espace de Fourier il est non-linéaire et également non-local. Il décrit des interactions dites triadiques entre ces trois nombres d'onde. Nous distinguons trois cas, ils sont illustrés dans la figure Fig.(2.2) :

- Si κ' et κ'' sont de même ordre de grandeurs : on parle d'interaction locale.
- Si $\kappa' > \kappa''$ ou inversement, on parlera d'interactions non-locales, tant que les différences ne sont pas trop élevées
- Si $\kappa' \gg \kappa''$ ou inversement, on parlera d'interactions distantes.

Le sens physique des interactions triadiques est le suivant : l'évolution au cours du temps d'une structure de taille donnée (associé au κ fixé) est altérée par l'injection de quantité de mouvement due à des structures de taille différentes.

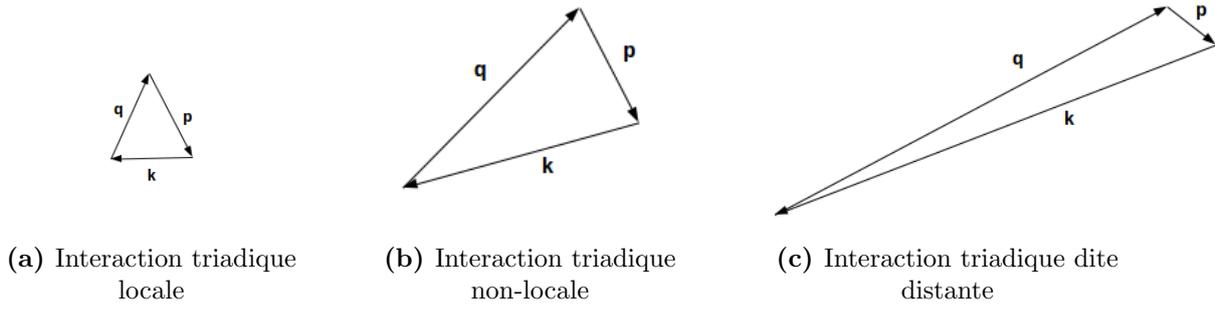


Figure 2.2 – Illustrations des interactions triadiques des vecteurs \mathbf{k} , \mathbf{p} et \mathbf{q} vérifiant $\mathbf{k} = \mathbf{p} + \mathbf{q}$.

De façon analogue à notre analyse, nous pouvons écrire la transformée de Fourier du tenseur des corrélations en deux points

$$\Phi_{ij}(\boldsymbol{\kappa}, \mathbf{x}, t) = \left(\frac{1}{2\pi}\right)^3 \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Gamma_{ij}(\mathbf{x}, \mathbf{r}, t) e^{i\boldsymbol{\kappa}\mathbf{r}} d\mathbf{r} \quad (2.2.16)$$

Enfin, pour $\mathbf{r} = \mathbf{0}$, et $i = j$, nous pouvons définir le spectre d'énergie cinétique $\hat{E}(\kappa, t)$:

$$\hat{E}(\kappa, t) = \frac{1}{2} \int_{S(\kappa)} \Phi_{ii}(\boldsymbol{\kappa}, t) dS(\boldsymbol{\kappa}) \quad (2.2.17)$$

C'est la somme de l'énergie portée par toutes les structures ayant un nombre d'onde $\boldsymbol{\kappa}$ incluses dans la sphère $S(\kappa)$ de rayon $\kappa = \|\boldsymbol{\kappa}\|$. Pour obtenir l'énergie cinétique totale portée par l'écoulement, il faut considérer les contributions provenant de tous les nombres d'ondes. Cela se fait en additionnant l'énergie contenue dans les coquilles sphériques (ou couronnes) d'épaisseur infinitésimale $d\kappa$ pour tous les modes à l'infini. On écrit donc :

$$k(t) = \int_{\mathbb{R}^+} \hat{E}(\kappa, t) d\kappa \quad (2.2.18)$$

Dans le cadre de la THI, le spectre $\hat{E}(\kappa, t)$ caractérise totalement le tenseur de corrélation. L'hypothèse d'isotropie est équivalente à écrire $\Phi_{ii}(\boldsymbol{\kappa}) = \Phi_{ii}(\kappa)$ ce qui permet d'obtenir une forme explicite du spectre :

$$\hat{E}(\kappa, t) = 2\pi\kappa^2 \Phi_{ii}(\kappa) \quad (2.2.19)$$

On peut relier ce spectre avec l'énergie cinétique turbulente dans l'espace physique :

$$k(t) = \int_{\mathbb{R}^+} 2\pi\kappa^2 \Phi_{ii}(\kappa) d\kappa = \frac{1}{2} \Gamma_{ii}(\mathbf{x}, 0, t) = \frac{1}{2} U_i^2(\mathbf{x}, t) \quad (2.2.20)$$

C'est une autre façon d'exprimer l'équivalence entre énergie véhiculée par les modes et énergie cinétique totale mesurée. Pour une simulation de taille 256^3 , on trace le spectre d'énergie figure Fig.(2.3). Cette figure illustre la cascade de Richardson/Kolmogorov. On y distingue nettement les vecteurs d'ondes responsables de l'injection d'énergie cinétique dans l'écoulement, ce sont les grandes échelles de l'écoulement. Puis le spectre suit la pente prédite par Kolmogorov $k \sim k^{-5/3}$ dans un grand domaine de vecteurs d'onde. C'est la cascade énergétique. Enfin, la

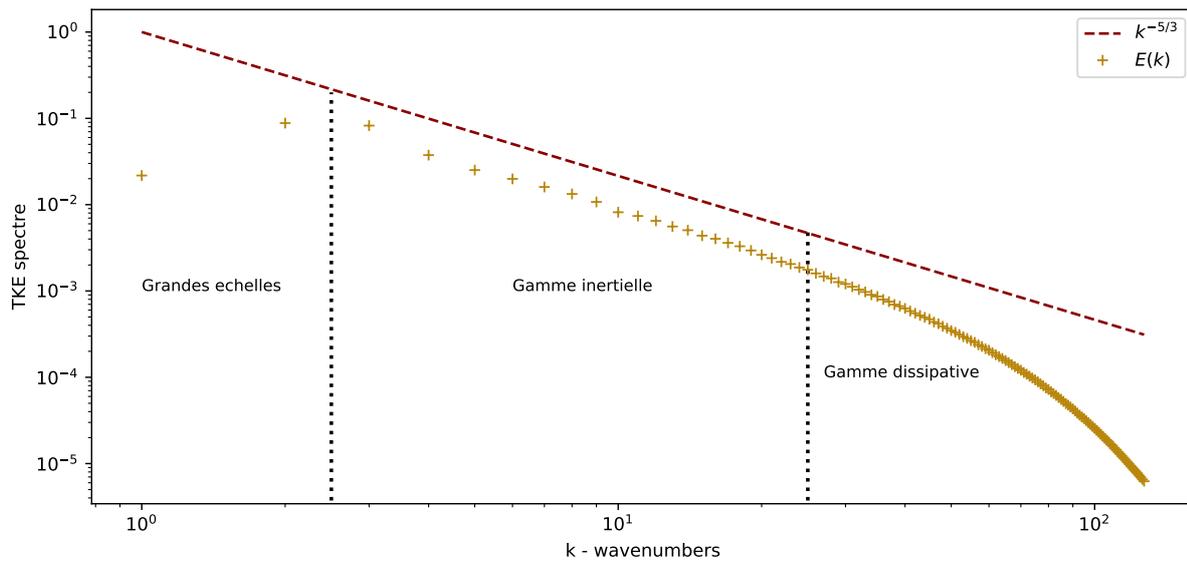


Figure 2.3 – Spectre d'énergie cinétique turbulente et identification des grandes échelles, et des gammes inertielle et dissipative. Une courbe de pente $k^{-5/3}$ est tracée en référence selon la théorie de Kolmogorov.

courbe change de comportement et la perte d'énergie d'une échelle à une autre plus petite est plus brutale. Ce sont les gammes dissipatives avec la plus petite échelle (le plus grand nombre d'onde) étant l'échelle de Komogorov.

Dans la majorité des cas, il nous est actuellement impossible de trouver une solution analytique à partir de l'équation de Navier-Stokes, les mécaniciens des fluides ont alors recours à des méthodes numériques pour y parvenir. Dans le cas général, la résolution numérique impose une discrétisation du domaine d'étude, et au plus le pas de discrétisation (Δ_x) est faible au plus la résolution sera élevée. Comme dans tout problème de discrétisation, il faut à tout prix éviter le sous-échantillonnage des phénomènes que l'on entend modéliser. Au vue de la taille de l'échelle de Kolmogorov Eq.(2.2.4), plus le Reynolds est élevé plus petite sera l'échelle de Kolmogorov. Ainsi, pour résoudre des écoulements au nombre de Reynolds élevé, il sera nécessaire de considérer un pas de discrétisation assez faible pour que l'échelle de Kolmogorov et les effets de dissipation soient pris en compte. Landau (Landau and Lifshitz (1987); Tran (2009)) estime également que le nombre de points de discrétisation nécessaires pour éviter ce sous-échantillonnage est de l'ordre de $Re^{9/4}$ d'où un coût de calcul de l'ordre de Re^3 (en secondes) (Pope, 2001). On voit donc qu'il semble difficile de monter en Reynolds pour le cadre d'une simulation directe (DNS) en particulier pour des applications industrielles, là où les résultats doivent être fiables et rapides. Le constat établi alors est le suivant : pour accélérer l'obtention de résultats tout en gardant un Reynolds élevé, il est inévitable de réduire la fiabilité de ces derniers à condition de pouvoir quantifier et contrôler l'erreur. C'est dans cette optique que des *modèles* de la turbulence ont été construits. Les deux grandes catégories de modèles se basent sur des réécritures des équations de Navier-Stokes : la première catégorie regroupe les modèles construits sur la décomposition de Reynolds des grandeurs turbulentes comme somme d'un champ moyen et d'un fluctuant. Dans cette catégorie, le champ moyen est résolu et les interactions entre partie fluctuante (non résolue) et champ moyen sont *modélisées* par

des *équations de fermeture*. La seconde catégorie consiste à filtrer les grandeurs turbulentes et à modéliser la physique dite sous-maille interagissant directement avec les champs filtrés. Cette catégorie est au cœur des travaux présentés dans cette thèse ; la section suivante est consacrée à la compréhension de cette approche.

2.3 Modélisation des grandes échelles et méthodes LES

Pour comprendre les simulations des grandes échelles (LES pour Large Eddy Simulation), il faut filtrer les équations de Navier-Stokes Eq.(2.1.10a) et Eq.(2.1.10b) c'est-à-dire convoluer un filtre passe-bas judicieusement choisi avec les équations sus-mentionnées.

2.3.1 Filtrage et problème de fermeture

Filtrage - Le nombre d'onde auquel le filtre agit est appelé le nombre d'onde de coupure κ_c , les phénomènes dont les nombres d'onde sont au delà ne sont pas résolus ; ils devront être modélisés. L'équivalent physique de cette coupure consiste à considérer une taille de grille $\bar{\Delta} = \pi/\kappa_c$ (voir Fig.(2.4)). Si on définit l'opération de filtrage d'une grandeur ξ avec un filtre noté G , on écrit $\bar{\xi}$ la quantité ξ filtrée et

$$\bar{\xi}_i = \int G(\mathbf{x} - \mathbf{x}') \xi_i(\mathbf{x}') d\mathbf{x}' \quad (2.3.1)$$

La transformée de Fourier de G se note \hat{G} c'est la fonction de transfert. Filtrer les plus grands modes revient à lisser les plus petites structures. Nous illustrons l'impact du filtrage sur le champ de vitesse figure Fig.(2.5). À droite, figure Fig.(2.5a) nous représentons le champ U_2 non

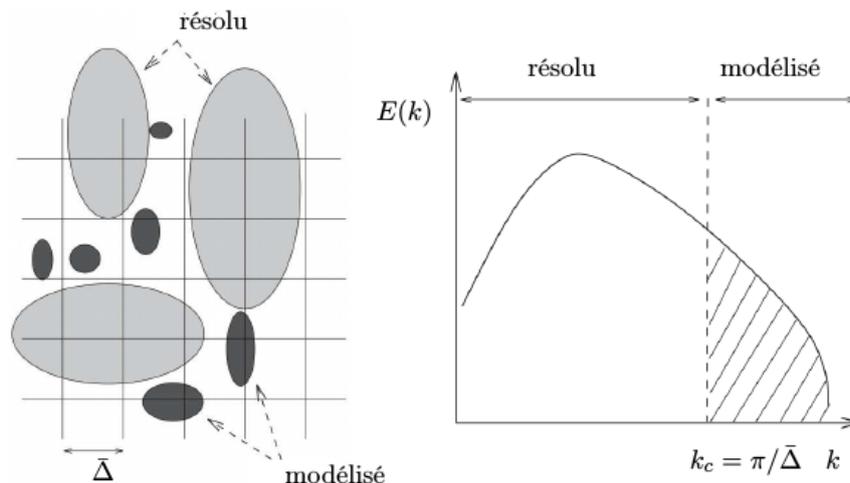
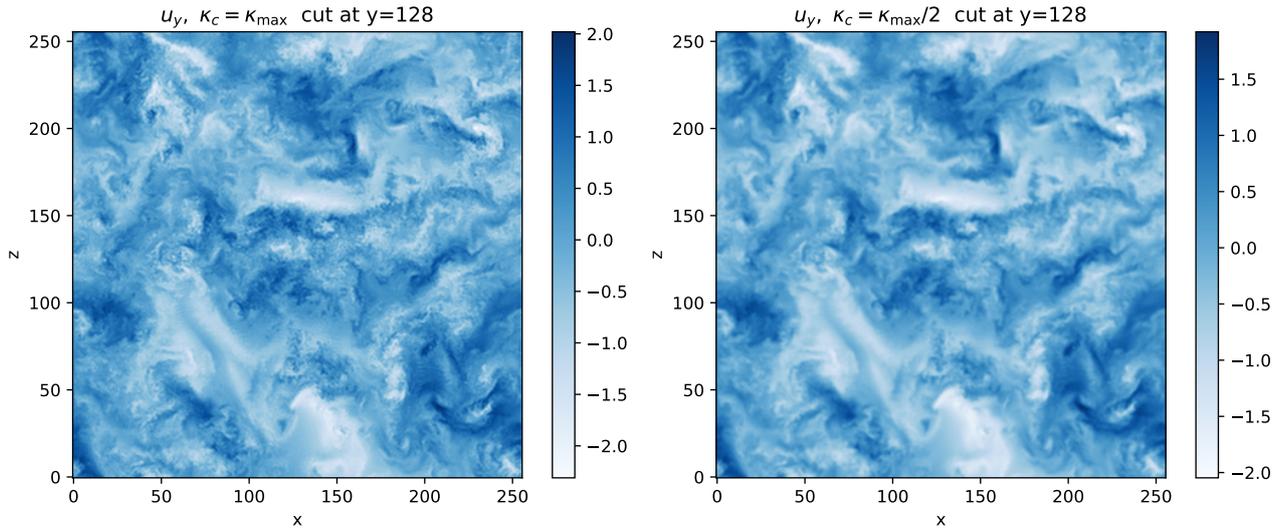


Figure 2.4 – Équivalence et interprétation physique (gauche) et spectrale (droite) du filtrage des grandeurs. Dans la figure de gauche les ovales foncées représentent les quantités (e.g. la vitesse) non résolues nécessitant un modèle. Idem pour la région hachurée figure de droite en terme de nombre d'onde et de spectre d'énergie (Vollant, 2015).



(a) Coupe de normale \vec{y} au centre du domaine du champ de vitesse U_y .

(b) Coupe de normale \vec{y} au centre du domaine du champ de vitesse U_y , filtrée à la fréquence $\kappa_c = K_{\max}/2$.

Figure 2.5 – Visualisation de l’effet du filtrage dans l’espace de Fourier. Les petites structures correspondant aux nombres d’onde supérieur à κ_c sont absentes après filtrage Fig.(2.5b).

filtré ; à gauche Fig.(2.5b) le même champ mais cette fois filtré à $\kappa_c = \kappa_{\max}/2$. On constate que le filtrage dans l’espace de Fourier se répercute par une absence des plus petites structures dans l’espace physique.

On applique l’opération de filtrage sur les équations de Navier-Stokes en utilisant la permutation dérivée-intégrale et la linéarité de l’opérateur intégrale :

$$\frac{\partial \overline{U}_i}{\partial t} + \frac{\partial \overline{U}_i \overline{U}_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \overline{P}}{\partial x_i} + \nu \frac{\partial^2 \overline{U}_i}{\partial x_j \partial x_j} \quad (2.3.2)$$

$$\frac{\partial \overline{U}_i}{\partial x_i} = 0 \quad (2.3.3)$$

Notons que le tenseur $\overline{U}_i \overline{U}_j$ est inconnu dans le cadre de la LES. En mécanique des fluides, il est commode de retrouver le terme de dérivée particulière à gauche du signe égal. Pour se faire on utilise une astuce consistant à faire apparaître des termes que l’on peut calculer dans la LES dans le terme de convection $\overline{U}_i \overline{U}_j = \overline{U_i U_j} + \overline{U_i} \overline{U_j} - \overline{U_i} \overline{U_j}$ ce qui permet de réécrire l’équation précédente :

$$\frac{\partial \overline{U}_i}{\partial t} + \overline{U_j} \frac{\partial \overline{U}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \overline{P}}{\partial x_i} + \nu \frac{\partial^2 \overline{U}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \quad (2.3.4)$$

avec

$$\tau_{ij} = \overline{U_i U_j} - \overline{U}_i \overline{U}_j \quad (2.3.5)$$

le tenseur de contraintes sous mailles. Cette équation est exacte dans la mesure où des variables ont été introduites mais aucune hypothèse n’a été faite (Ferziger and Reynolds, 1979). Toujours de manière exacte, on peut décomposer τ_{ij} en plusieurs termes selon les différentes échelles

résolues ou simulées (*ibid* et Leonard (1975)). On écrit alors

$$\tau_{ij} = L_{ij} + C_{ij} + R_{ij} \quad (2.3.6)$$

Décomposition de Leonard - Ces trois tenseurs contiennent les trois différentes interactions entre échelles inconnues. Le premier L_{ij} appelé tenseur des contraintes de Leonard s'écrit

$$L_{ij} = \overline{\overline{U_i U_j}} - \overline{U_i} \overline{U_j} \quad (2.3.7)$$

Il décrit les interactions dont les éléments du triplet $(\kappa, \kappa', \kappa'')$ vérifient les conditions :

$$\begin{aligned} \kappa_c &< \kappa < 2\kappa_c \\ \kappa', \kappa'' &< \kappa_c \end{aligned}$$

Ces interactions font intervenir des grandes échelles (les échelles résolues) de l'ordre de κ_c Volland (2015). Elles donnent naissance à une partie des échelles non résolues (Gatski et al., 1996). Le tenseur C_{ij} décrit les interactions croisées entre échelles (ou modes) résolues et non résolues. Il fait intervenir la vitesse résiduelle post-filtrage $u'_i = U_i - \overline{U_i}$; il s'écrit

$$C_{ij} = \overline{\overline{U_i u'_j}} + \overline{u'_i U_j} \quad (2.3.8)$$

Il décrit les interactions les éléments du triplet $(\kappa, \kappa', \kappa'')$ vérifient les conditions :

$$\begin{aligned} \kappa &< \kappa_c \\ \kappa_c &< \max(\kappa', \kappa'') < 2\kappa_c \\ \min(\kappa', \kappa'') &< \kappa_c \end{aligned}$$

Enfin, le tenseur de Reynolds sous-maille R_{ij} décrit les interactions entre les petites structures aux nombres d'onde au delà de κ_c ; il fait donc intervenir uniquement les vitesses résiduelles :

$$R_{ij} = \overline{u'_i u'_j} \quad (2.3.9)$$

De par leur construction et leur nature, ces termes ne sont pas calculables dans une simulation aux grandes échelles, c'est ici que va intervenir les premières hypothèses pour établir des modèles permettant d'avoir une idée de l'activité sous mailles et sa conséquence sur l'évolution des quantités turbulentes.

Dans une logique de modélisation, il est nécessaire de mettre au point un modèle dont les termes sous mailles respectent le plus possible les transferts énergétiques d'une échelle à une autre. Il faut dans un premier temps identifier l'implication des termes sous mailles au sein des processus de transferts. On multiplie alors l'équation Eq.(2.3.4) par $\overline{U_j}$. On obtient l'équation d'énergie cinétique turbulente $K(\mathbf{x}, t) = \Gamma_{ii}(\mathbf{x}, t)/2$:

$$\frac{DK}{Dt} = -\frac{\partial \overline{U_i \bar{p}}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\frac{\nu}{2} \frac{\partial (\overline{U_i U_i})}{\partial x_j} - \overline{U_i} \tau_{ij} \right] - \nu \frac{\partial \overline{U_i}}{\partial x_j} \frac{\partial \overline{U_i}}{\partial x_j} - (-\tau_{ij} \overline{S_{ij}}) \quad (2.3.10)$$

avec \bar{p} la pression dynamique filtrée. Cette équation met en évidence l'existence de plusieurs mécanismes de transfert d'énergie cinétique :

- $\nabla \cdot \bar{U}\bar{p}$ le transfert d'énergie cinétique par des effets de pression.
- Le terme $\nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}) = \mathbf{U} \cdot (\nabla \cdot \boldsymbol{\tau})$ représente la diffusion de l'énergie cinétique par interactions entre échelles résolues non résolues.
- Le terme $\Pi = -\boldsymbol{\tau} : \bar{\mathbf{S}}$ représente la dissipation de l'énergie cinétique par les échelles sous mailles.
- Les termes $\nu \nabla \cdot \nabla K$ et $\nu (\nabla \bar{\mathbf{U}})^2$ représentent respectivement la diffusion de l'énergie cinétique par effets visqueux et la destruction de l'énergie cinétique par ces mêmes effets.

Dans notre cas, le terme mettant en jeu le champ de pression s'annule par incompressibilité du champ de vitesse. Les termes de dissipation font intervenir les échelles résolues. Le terme Π caractérise le transfert d'énergie des grandes échelles vers les petites échelles lorsqu'il est positif, mais également le transfert inverse appelé la rétrodiffusion ou *backscattering*. Ce phénomène traduit l'injection d'énergie depuis les petites échelles vers les grandes ; c'est un phénomène plutôt rare. Le produit Π est parfois appelé dissipation effective dans le sens où du point de vue des grandes échelles, il représente généralement un puits (Borue and Orszag, 1998). Nous l'avons mentionné lorsque ce terme est négatif le sens physique est opposé et agira comme une source d'énergie par rétrodiffusion. Les termes en τ_{ij} dans l'équation Eq.(2.3.10) sont des éléments inconnus. De manière générale, lorsqu'un système d'équations fait intervenir plus d'inconnues que de relations, nous essayons d'établir des équations supplémentaires pour fermer le système. Or dans le cas des méthodes LES, dès lors que l'on cherche à modéliser le terme sous-mailles, toutes les tentatives algébriques se sont avérées être déjouées par l'apparition de terme eux mêmes inconnus, nécessitant à leur tour d'autres équations et ainsi de suite⁵ ; on parlera de problème de fermeture.

2.3.2 Modèles de Smagorinsky et du Gradient

Il existe énormément de modèles LES pour fermer le problème, voir par exemple (Sagaut, 2006) pour une explication détaillée de chaque modèle. Dans cette section nous présentons brièvement deux des modèles les plus répandus dans les LES à savoir le Smagorinsky (Smagorinsky, 1963) et le modèle du gradient ou modèle de Clark (Clark et al., 1979). Ces deux modèles sont de prime importance puisqu'ils seront utilisés comme étalons de mesures des performances des modèles augmentés présentés chapitre 4 et utilisés chapitre 5. Les deux modèles de la littérature seront utilisés dans leur forme classique, non dynamique.

Modèle de Smagorinsky - Le modèle de Smagorinsky est un modèle dit fonctionnel qui cherche à reproduire les transferts énergétiques entre les échelles résolues et les échelles modélisées (Sagaut, 2006; Vollant, 2015). Il se base sur l'hypothèse d'une viscosité sous mailles ν_s analogue à la diffusion moléculaire. On suppose alors que les transferts d'énergie sous mailles sont

5. Comme dans de nombreux cas on remarque que la cascade énergétique trouve son analogue dans la cascade d'équations nécessaire pour la modéliser.

entièrement pilotés par cette diffusion. L'équation de fermeture du modèle Smagorinsky s'appuie en quelque sorte sur la formulation de Boussinesq

$$\tau_{ij}^{Smago} = -2\nu_s \overline{S_{ij}} + \frac{1}{3} \tau_{kk} \delta_{ij} \quad (2.3.11)$$

avec $\overline{S_{ij}}$ le tenseur de déformation filtrée défini Eq.(2.1.8) et ν_s la viscosité sous-maillages définie par

$$\nu_s = C_s \bar{\Delta} \|\overline{S_{ij}}\| \quad (2.3.12)$$

C_s le coefficient de Smagorinsky défini par le coefficient de Kolmogorov, généralement $C_s \approx 0.17$ valeur issue de la représentation en loi de puissance du spectre d'énergie par Schumann (1991), plus de détails peuvent être trouvés entre autres dans Vreman et al. (1997). Le modèle tel quel est connu pour être trop dissipatif dans les régions laminaires de l'écoulement et le tenseur des contraintes sous-maillages résultants est peu corrélé avec le tenseur issu de la DNS filtrée (FDNS par la suite). Néanmoins, il reste un des modèles les plus utilisés dans l'industrie. Il est considéré viable dès lors que l'écoulement est assez turbulent, comme nous le verrons chapitre 5.

Modèle du gradient - Le modèle du gradient ou de Clark (Clark et al., 1979) est un modèle dit structurel dont le but est de reproduire avec le plus d'exactitude possible les composantes du tenseur sous-maillages τ_{ij} (Sagaut, 2006; Vollant, 2015). Il est basé sur la décomposition du tenseur des contraintes faite par Leonard (1975) et le modèle de Similarité proposé par Bardina et al. (1984), revisité par Liu et al. (1994). Principalement ce remaniement est dû au fait que Clark et al. (1979) ont utilisé les développements de Taylor sur la vitesse résiduelle qui ne connaît pas de variations sur les puissances de $\bar{\Delta}$ assez lisses (Love, 1980; Vreman et al., 1996). Dans le développement utilisé par Liu et al. (1994) qui est la version que nous utilisons, le développement se base sur le modèle de similarité qui écrit

$$\tau_{ij}^B = \overline{\overline{U_i U_j}} - \overline{\overline{U_i}} \overline{\overline{U_j}} \quad (2.3.13)$$

En injectant les deux quantités suivantes

$$\overline{\overline{U_i U_j}} = \overline{U_i U_j} + \frac{\bar{\Delta}}{24} \frac{\partial^2 \overline{U_i U_j}}{\partial x_k \partial x_k} + \mathcal{O}(\bar{\Delta}^4) \quad (2.3.14)$$

$$\overline{\overline{U_i}} = \overline{U_i} + \frac{\bar{\Delta}}{24} \frac{\partial^2 \overline{U_i}}{\partial x_k \partial x_k} + \mathcal{O}(\bar{\Delta}^4) \quad (2.3.15)$$

On obtient l'expression du tenseur des contraintes du modèle du gradient

$$\tau_{ij} = \frac{\bar{\Delta}}{12} \frac{\partial \overline{U_i}}{\partial x_k} \frac{\partial \overline{U_j}}{\partial x_k} \quad (2.3.16)$$

La principale raison de l'utilisation de ce modèle est sa performance. Des articles assez récents (par exemple Xie et al. (2019); Yuan et al. (2020); Zhou et al. (2019)) montrent des corrélations de plus de 90% entre les composantes de τ_{ij} prédites par le modèle du gradient et la FDNS. De plus, contrairement aux formes de Smagorinsky dynamiques ainsi qu'au modèle de similarité initial, le modèle de Clark ne nécessite pas d'opérations de filtrage supplémentaires et sa construction ne nécessite pas de calculs supplémentaires car les dérivées qu'il nécessite font par-

tie intégrante du corps du solver même. Il subsiste malgré tout des limites à l'utilisation de ce modèle tel qu'il est présenté ici. La principale étant le fait que la rétrodiffusion est surévaluée, et de manière générale que le modèle dissipe trop (il est même parfois instable voir l'étude [Vreman et al. \(1996\)](#)). Mais dans la mesure où nous pouvons identifier les limites de ce modèle, nous pouvons avoir un regard plus critique sur les résultats des modèles augmentés par l'IA présentés par la suite. Et dans un même temps, les très bonnes performances sur la modélisation de τ de ce modèle en font un objectif à dépasser.

Pour résumer, nous utiliserons les deux modèles comme outils d'étalonnage de nos modèles. Les modélisations du modèle de Smagorinsky ne seront pas constamment détaillées puisqu'il est moins fiable que le modèle de Clark. Ses sorties seront malgré tout considérées pour deux raisons : dans un premier temps car il est encore beaucoup utilisé, et dans un deuxième temps pour situer les performances des réseaux de neurones à la base des modèles que nous présenterons chapitre 5. Nous le répétons ici, le modèle du gradient constitue le véritable défi notamment tant au niveau du tenseur lui-même qu'aux niveaux *vectoriel* et *scalaire* que nous définissons dans la section suivante.

2.4 Évaluation des modèles : outils d'analyse

2.4.1 Grandeurs à évaluer

Niveaux tensoriel, vectoriel et scalaire - La majorité des travaux effectués sur les modèles LES, qu'ils soient à base d'intelligence artificielle ou non, comparent les PDF ou les corrélations entre les composantes modélisées et cibles du tenseur de contraintes ; et également les transferts d'énergie entre grandes et petites échelles. D'après la terminologie de ([Ferziger and Reynolds, 1979](#)), il existe trois niveaux de comparaisons, et chacun exprime un impact différent du tenseur sous-mailles modélisé sur les autres grandeurs principales de la Turbulence :

- 1) Niveau tensoriel : il s'agit d'évaluer la correspondance entre les composantes du tenseur modélisé et celles du tenseur attendu. À ce niveau on considère le tenseur τ_{ij} .
- 2) Niveau vectoriel : il s'agit d'évaluer l'impact de la prédiction dans le devenir de la quantité de mouvement. Cela équivaut à l'accélération induite par le vecteur $\partial\tau_{ij}/\partial x_j$ qui apparaît dans l'équation Eq.(2.2.15a). Ce terme marque la contribution du tenseur τ_{ij} dans les transferts d'énergie sans perte (diffusion).
- 3) Niveau scalaire : il s'agit d'évaluer l'impact de la modélisation d'un point de vue énergétique au travers les grandeurs $-\bar{S}_{ij}\tau_{ij}$ et $u_i\partial\tau_{ij}/\partial x_j$.

Au niveau scalaire, nous considérerons majoritairement la grandeur $\Pi = -\tau_{ij}\bar{S}_{ij}$ comme dans la majorité des articles à notre connaissance ; en revanche, nous mentionnons le terme $u_i\partial\tau_{ij}/\partial x_j$ puisqu'il apparaît explicitement dans l'équation Eq.(2.3.10).

Au regard des différentes variables considérées, nous remarquons que l'information contenue dans les niveaux vectoriel et scalaire est contractée (au sens tensoriel) et la nécessité d'évaluer les correspondances à ces niveaux peut donc paraître superflue. Néanmoins, la philosophie des modèles LES étant de "supprimer" de l'énergie localement, il semble donc que la correction engendrée par le tenseur des contraintes et son impact d'un point de vue énergétique soit in-

dispensable, d'où l'évaluation au niveau scalaire. Le même argument peut être utilisé pour le niveau vectoriel, d'un point de vue dynamique de l'écoulement.

Différentes contraintes et décomposition de Leonard - Rappelons l'équation Eq.(2.3.6) :

$$\tau_{ij} = L_{ij} + C_{ij} + R_{ij} \quad (2.3.6)$$

Elle met en exergue trois différents niveaux d'interactions. Nous y avons identifié L_{ij} caractérisant les interactions entre échelles résolues dont la résultante appartient au domaine sous-maillles, C_{ij} décrivant des interactions entre échelles résolues et non résolues susceptibles de mettre en jeu le principe de rétrodiffusion. La direction globale du transfert d'énergie en jeu dans ce tenseur est dans le sens : échelles résolues aux non résolues. Enfin, le troisième terme R_{ij} appelé le véritable tenseur d'interactions sous mailles, décrit les interactions des échelles non résolues entre elles occasionnant la majorité des actions sous-maillles sur les champs résolus (Gatski et al., 1996). D'après nos discussions 2.3.2, il apparaît que les limites du performant modèle de Clark proviennent de la mauvaise estimation de la rétrodiffusion principalement décrite par le tenseur R_{ij} .

Pour les modèles LES, il est impossible de calculer les termes L_{ij} , C_{ij} et R_{ij} notamment puisqu'ils mettent en jeu la vitesse résiduelle post-filtrage⁶. Néanmoins nous pouvons construire le manque de précision de chacun des termes séparément avec la formulation suivante :

$$L_{ij}^{\text{model}} = \tau_{ij}^{\text{model}} - C_{ij}^{\text{FDNS}} - R_{ij}^{\text{FDNS}} \quad (2.4.1)$$

$$C_{ij}^{\text{model}} = \tau_{ij}^{\text{model}} - L_{ij}^{\text{FDNS}} - R_{ij}^{\text{FDNS}} \quad (2.4.2)$$

$$R_{ij}^{\text{model}} = \tau_{ij}^{\text{model}} - L_{ij}^{\text{FDNS}} - C_{ij}^{\text{FDNS}} \quad (2.4.3)$$

Intuitivement cette façon de procéder revient à quantifier l'écart à la DNS causé par la modélisation sur chacune des composantes sous mailles indépendamment les unes des autres. L'exposant "model" se veut générique ; ces trois formulations sont également utilisables pour les composantes sous-maillles du tenseur prédit par IA⁷. Notons que

$$\tau_{ij}^{\text{model}} \neq L_{ij}^{\text{model}} + C_{ij}^{\text{model}} + R_{ij}^{\text{model}} \quad (2.4.4)$$

De même, nous pouvons définir ces quantités d'un point de vue dissipatif en multipliant chacune des équations (2.4.1), (2.4.2) et (2.4.3) par $\overline{S}_{ij}^{\text{FDNS}}$. Nous obtenons alors le manque de précision de la dissipation propre à chacun des tenseurs sous-maillles :

$$\Pi_{L_{ij}}^{\text{model}} = \Pi^{\text{model}} - \Pi_{C_{ij}}^{\text{FDNS}} - \Pi_{R_{ij}}^{\text{FDNS}} \quad (2.4.5)$$

$$\Pi_{C_{ij}}^{\text{model}} = \Pi^{\text{model}} - \Pi_{L_{ij}}^{\text{FDNS}} - \Pi_{R_{ij}}^{\text{FDNS}} \quad (2.4.6)$$

$$\Pi_{R_{ij}}^{\text{model}} = \Pi^{\text{model}} - \Pi_{L_{ij}}^{\text{FDNS}} - \Pi_{C_{ij}}^{\text{FDNS}} \quad (2.4.7)$$

6. Pour L_{ij} dans la mesure où un des nombre d'onde du triplet est plus élevé que le nombre d'onde de coupure, l'information nécessaire pour l'évaluer est manquante.

7. Nous précisons qu'il ne s'agit pas de calculer les composantes sous mailles mais d'en mesurer l'écart imposé par la modélisation.

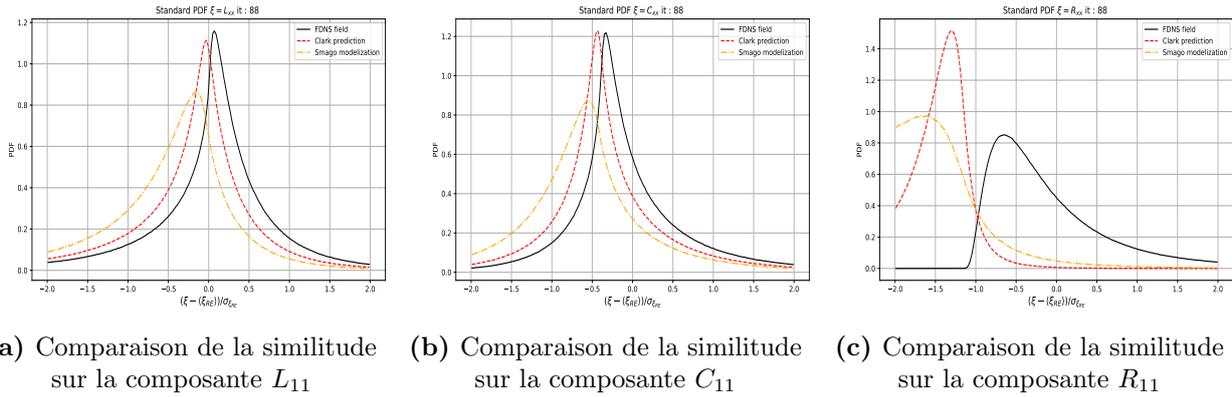


Figure 2.6 – Pour une simulation dont le Reynolds à l’échelle de Taylor est $Re_\lambda \approx 240$, nous comparons la première composante des trois tenseurs de Eq.(2.3.6). Les courbes brisées rouge et orange représentent la modélisation de ces grandeurs par le modèle de Clark, et de Smagorinsky. Les courbes noires représentent les grandeurs issues de la DNS filtrée.

Calculer ces grandeurs permet d’identifier les manquements des modèles ; et comparer ces écarts entre les modèles algébriques et les modèles augmentés par IA rend possible l’évaluation de la qualité de la modélisation par prédiction des interactions sous mailles d’un point de vue physique.

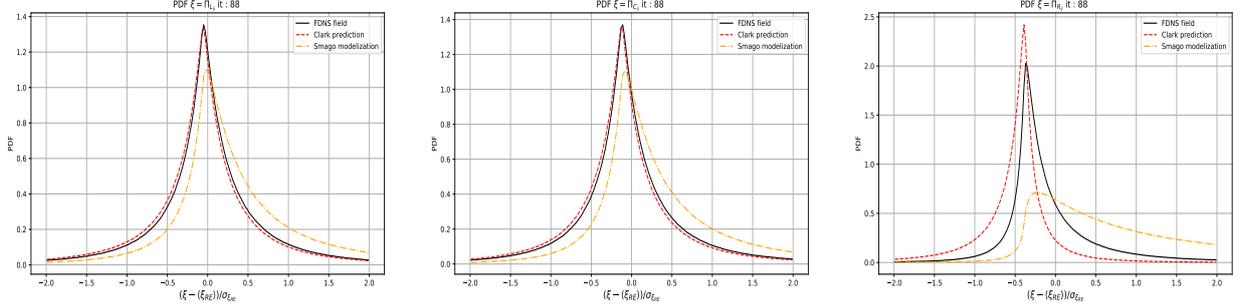
2.4.2 Outils pour évaluer la qualité des champs modélisés

Utilisation des PDF - La PDF (probability density function ou fonction de densité de probabilité) permet de connaître la probabilité d’occurrence des amplitudes (ou valeurs) du signal étudié. Cette représentation est particulièrement adéquate quand il s’agit de comparer des champs 3D tout en ayant une visualisation claire de la qualité de la prédiction/modélisation. L’utilisation des PDF est classique dans la littérature y compris quand il s’agit d’évaluer les modèles augmentés. Cette représentation des données permet d’avoir des informations supplémentaires comme l’intermittence que l’on peut retrouver dans les événements rares.

Par exemple, nous traçons les composantes du tenseur τ_{ij} de l’équation Eq.(2.3.6) pour les modèles du Gradient (ligne brisée rouge) et de Smagorinsky (ligne brisée orange). Les trois grandeurs L_{ij} , C_{ij} et R_{ij} sont calculées utilisant les formules Eqs.(2.4.1), (2.4.2) et (2.4.3) pour chacun des deux modèles. La ligne continue noire représente la valeur de ces grandeurs calculées par les formules exactes Eqs.(2.3.7), (2.3.8) et (2.3.9). Dans les trois figures précédentes, les PDF des grandeurs L_{11} , C_{11} et R_{11} illustrent bien les forces et les faiblesses des modèles du Gradient et de Smagorinsky. Pour les interactions impliquant les échelles résolues figures Figs.(2.6a) et (2.6b), les modélisations sont relativement bonnes en particulier celles du modèle du Gradient. Par contre, pour le terme impliquant uniquement les échelles sous mailles Fig.(2.6c), on remarque que les modèles sont moins précis et que les PDF des modèles et de la DNS filtrée n’ont quasiment pas de similitude.

Nous pouvons également illustrer l’impact estimé de ces modélisations au niveau de la diffusion utilisant cette fois les formules Eqs.(2.4.5), (2.4.6) et (2.4.7) pour les modèles, et leur équivalent en considérant τ_{ij} issue de la DNS filtrée. La contraction des composantes τ_{ij} modélisées et du $\overline{S_{ij}}^{\text{DNS}}$ fait que les erreurs s’annulent mutuellement diminuant l’impact énergétique des er-

reurs constatées. Les transferts énergétiques dus aux composantes du tenseur R_{ij} constituent la difficulté majeure des modèles algébriques présentés ici.



(a) Comparaison de la similitude sur la composante $\Pi_{L_{ij}}$ (b) Comparaison de la similitude sur la composante $\Pi_{C_{ij}}$ (c) Comparaison de la similitude sur la composante $\Pi_{R_{ij}}$

Figure 2.7 – Pour les trois graphiques, la courbe brisée rouge et orange représentent les modélisations de ces grandeurs par le modèle de Clark, et Smagorinsky (respectivement). Les courbes noires représentent les grandeurs issues de la DNS filtrée.

Scores : Pearson, \mathcal{R}^2 et erreur relative - Que ce soit pour les modèles algébriques ou augmentés, il est primordial de donner une mesure de l'erreur que ces modélisations commettent par rapport aux données de référence issues de DNS ou d'expériences. Considérant le caractère aléatoire de la turbulence, il est également nécessaire d'utiliser des outils statistiques. Une métrique qui apparaît souvent dans la littérature est le coefficient de corrélation entre les grandeurs modélisées et les cibles. Cette corrélation est basée sur les premier et deuxième moments des deux champs comparés (Pope, 2001). On définit ce coefficient de corrélation (C.C) entre un champ turbulent cible u^{Re} et son correspondant modélisé u^{Model} par :

$$CC(u) = \frac{\left\langle \left(u^{\text{Re}} - \langle u^{\text{Re}} \rangle_{x,y,z} \right) \left(u^{\text{Model}} - \langle u^{\text{Model}} \rangle_{x,y,z} \right) \right\rangle_{x,y,z}}{\left[\left\langle \left(u^{\text{Re}} - \langle u^{\text{Re}} \rangle_{x,y,z} \right)^2 \right\rangle_{x,y,z} \right]^{1/2} \left[\left\langle \left(u^{\text{Model}} - \langle u^{\text{Model}} \rangle_{x,y,z} \right)^2 \right\rangle_{x,y,z} \right]^{1/2}} \quad (2.4.8)$$

La forme utilisée dans l'équation précédente est une forme discrète, où la notation $\langle \Lambda \rangle_{x,y,z}$ désigne l'opérateur de moyenne sur l'ensemble des points de l'espace :

$$\langle \Lambda \rangle_{x,y,z} = \frac{1}{N_x} \frac{1}{N_y} \frac{1}{N_z} \sum_{i_x=0}^{N_x-1} \sum_{i_y=0}^{N_y-1} \sum_{i_z=0}^{N_z-1} \Lambda_{i_x, i_y, i_z} \quad (2.4.9)$$

Dans nos travaux, nous utilisons les C.Cs dans lesquels les moyennes sont calculées selon chacun des axes de l'espace. Par exemple le C.C selon l'axe x , la formule Eq.(2.4.8) devient alors :

$$CC(u; x) = \frac{\left\langle \left(u^{\text{Re}} - \langle u^{\text{Re}} \rangle_{y,z} \right) \left(u^{\text{Model}} - \langle u^{\text{Model}} \rangle_{y,z} \right) \right\rangle_{y,z}}{\left[\left\langle \left(u^{\text{Re}} - \langle u^{\text{Re}} \rangle_{y,z} \right)^2 \right\rangle_{y,z} \right]^{1/2} \left[\left\langle \left(u^{\text{Model}} - \langle u^{\text{Model}} \rangle_{y,z} \right)^2 \right\rangle_{y,z} \right]^{1/2}} \quad (2.4.10)$$

Dans ce cas l'opérateur de moyenne $\langle \Lambda \rangle_{y,z}$ s'écrira

$$\langle \Lambda \rangle_{y,z} = \frac{1}{N_y} \frac{1}{N_z} \sum_{i_y=0}^{N_y-1} \sum_{i_z=0}^{N_z-1} \Lambda_{i_y, i_z} \quad (2.4.11)$$

C'est un vecteur qui permettra de visualiser l'évolution moyenne selon les points de discrétisation dans la direction \vec{x} . De même nous définissons $CC(u; y)$ et $CC(u; z)$.

En statistique, cette corrélation (sur l'ensemble du domaine) est appelée la corrélation de Pearson. On peut montrer que ce coefficient est la racine carrée d'une autre grandeur très importante dans l'évaluation d'une régression dans le sens général : le score \mathcal{R}^2 . On définit \mathcal{R}^2 également indépendamment de la définition donnée Eq.(2.4.8) :

$$\mathcal{R}^2(u) = 1 - \frac{\left[\left\langle \left(u^{\text{RE}} - \langle u^{\text{Model}} \rangle_{x,y,z} \right) \right\rangle_{x,y,z} \right]^2}{\left[\left\langle \left(u^{\text{RE}} - \langle u^{\text{RE}} \rangle_{x,y,z} \right) \right\rangle_{x,y,z} \right]^2} \quad (2.4.12)$$

Ce score mesure la proportion de variance du champ cible recouverte dans la prédiction. On peut le voir également comme une évaluation de l'information sur la variance manquante. Si par exemple nous mesurons $\mathcal{R}^2(\tau_{xx}^{\text{ML}}) = 0.94$, l'interprétation de ce score sera "le champ τ_{xx}^{ML} prend en compte 94% de la variance du champ de référence correspondant τ_{xx}^{DNS} ".

Si le \mathcal{R}^2 est peu utilisé dans la littérature relative à la turbulence (Singh et al. (2017a) ou Pan and Duraisamy (2018)), le coefficient de corrélation défini comme la corrélation de Pearson est abondamment utilisé dans la littérature classique comme (Pope, 2001), (George, 2013) ou (Borue and Orszag, 1998) et également dans les articles les plus récents comme (Beck et al., 2019; Xie et al., 2019; Yuan et al., 2020; Zhou et al., 2019). Dans ces deux dernières occurrences, les auteurs utilisent également l'erreur relative notée E_r pour quantifier d'une autre façon les différences entre champs cibles et prédits. En utilisant les mêmes notations que Eq.(2.4.8) et Eq.(2.4.10) respectivement

$$E_r(u) = \frac{\left[\left\langle \left(u^{\text{Re}} - u^{\text{Model}} \right)^2 \right\rangle_{x,y,z} \right]^{1/2}}{\left[\langle u^{\text{Re}} \rangle_{x,y,z} \right]^{1/2}} \quad (2.4.13)$$

$$E_r(u; x) = \frac{\left[\left\langle \left(u^{\text{Re}} - u^{\text{Model}} \right)^2 \right\rangle_{y,z} \right]^{1/2}}{\left[\langle u^{\text{Re}} \rangle_{y,z} \right]^{1/2}} \quad (2.4.14)$$

Améliorer les modélisations existantes - Calculons les trois scores C.C, R^2 et E_r sur l'ensemble du volume pour les grandeurs tracées figures Figs.(2.7a), (2.7b) et (2.7c). Nous renseignons les résultats dans le tableau Tab.(2.1).

La présence de scores négatifs dans les différentes mesures renseignées Tab.(2.1) illustre la

Modèle LES	Pearson	\mathcal{R}^2	E_r
Clark	0.97/0.96/-0.57	0.92/0.92/-1.56	0.28/0.28/1.42
Smagorinsky	0.78/0.75/0.39	-0.13/-0.15/-34	1.1/1.1/5.6

Table 2.1 – Scores sur Π_{Lij}^{model} , Π_{Cij}^{model} et Π_{Rij}^{model} pour les deux modèles.

mauvaise qualité des évaluations de la dissipation par les échelles modélisées. Pour le modèle de Clark, les scores sont toutefois très bons lorsqu’il s’agit des grandeurs Π_{Lij}^{model} et Π_{Cij}^{model} ; les corrélations sont cependant mauvaises sur Π_{Rij}^{model} .

Un mot sur le Reynolds à l’échelle de Taylor - Avant de conclure cette section, il est nécessaire de définir le nombre de Reynolds à l’échelle dite de Taylor. De manière surprenante, Pope (2001) ne lui donne pas d’interprétation bien définie et pourtant elle est communément utilisée en particulier pour décrire la turbulence de grilles. Cette échelle est liée à l’échelle de Kolmogorov. Le Reynolds à cette échelle est donnée par

$$Re_\lambda = \frac{u^{\text{RMS}} \lambda}{\nu} \quad (2.4.15)$$

avec

$$u^{\text{RMS}} = \sqrt{\frac{\langle u_i u_i \rangle}{3}} \quad \text{et} \quad \lambda = \sqrt{\frac{10\nu u^{\text{RMS}2}}{\langle \varepsilon \rangle}} \quad (2.4.16)$$

λ définie de la sorte est appelée l’échelle de Taylor. De façon analogue aux équations Eq.(2.2.4), nous pouvons définir Re_λ en fonction du Reynolds aux grandes échelles :

$$Re_\lambda = \sqrt{\frac{20}{3}} Re_L \quad (2.4.17)$$

C’est donc une échelle intermédiaire.

Conclusion du chapitre - Toute l’idée de ce manuscrit est de construire une intelligence artificielle (IA) capable de prédire des composantes des contraintes sous-maïlles dont la qualité aux niveaux tensoriel, vectoriel et scalaire est au plus proche des composantes correspondantes issues de la DNS. C’est au travers des PDF et des différents scores présentés plus haut que ces améliorations seront mesurées. Un défi supplémentaire est l’analyse des interactions constituant le tenseur τ_{ij} . Nous chercherons par exemple à déterminer si les modélisations par IA réussissent à recouvrir plus d’information sur la réalité sous mailles afin d’estimer *a priori* l’impact de modèles LES construits autour de ces intelligences artificielles sur le court terme voire le moyen terme.

Le chapitre suivant pose les bases des réseaux de neurones et plus particulièrement des réseaux de neurones à convolutions. Nous avons également présenté les apports et les utilisations inédites de notre travail au niveau des briques élémentaires du réseau.

Chapitre 3

Apprentissage automatique par réseaux de neurones artificiels

L'intelligence artificielle et plus particulièrement les réseaux de neurones sont au cœur des travaux effectués durant cette thèse. Après l'introduction des principes fondamentaux des réseaux de neurones en général, et des réseaux à convolutions en particulier section 3.1, nous proposons une revue des principaux *hyper-paramètres* dont l'exploration a été décisive pour l'obtention de nos résultats principaux. Par hyper-paramètres, l'on désigne généralement les caractéristiques liées à l'architecture même du réseau comme sa profondeur, ses non-linéarités, son objectif - sa fonction de coût - ou bien les ajustements possibles relatifs à son apprentissage : stratégie d'optimisation, taux d'apprentissage (ou *learning rate*), stratégie d'initialisation des paramètres et d'autres. Les non linéarités ou fonctions d'activation seront discutées section 3.2, à défaut d'en proposer une inédite, nous apporterons des arguments sur les performances d'activations plus ou moins récentes et peu utilisées dans la littérature. Nous aborderons également l'objectif de l'apprentissage à travers la définition de la fonction de coût 3.3 en insistant sur la notion d'alignement. Dans cette même section, nous discuterons des méthodes d'optimisation et les raisons qui font qu'une méthode est plus performante qu'une autre. Ici aussi, notre exploration a révélé que l'optimiseur le plus utilisé dans tous les domaines confondus, y compris ceux de l'IA dans la turbulence, n'est pas d'emblée le plus optimal. Cette section est complétée de l'annexe A dans laquelle nous détaillons la notion de moment et son impact sur l'apprentissage.

3.1 Introduction aux réseaux de neurones

3.1.1 Fonctionnement et apprentissage d'un réseau de neurones

Un réseau de neurones est un ensemble de couches de *neurones* reliées entre elles par des *paramètres*, associant à chaque entrée X une image $\mathcal{F}(X)$. Les couches de neurones intermédiaires sont appelées les couches cachées. Chacune de ses couches est constituée de plusieurs unités de calculs appelées neurones dont le but est de *résumer* puis *transformer* le signal reçu. Les couches de neurones travaillent de concert dans le but de *minimiser l'erreur commise par l'ensemble* c'est-à-dire la *prédiction* du réseau, par rapport aux *objectifs* ou cibles de *l'apprentissage*.

Le but de cette section est double : dans un premier temps nous introduisons les concepts fonda-

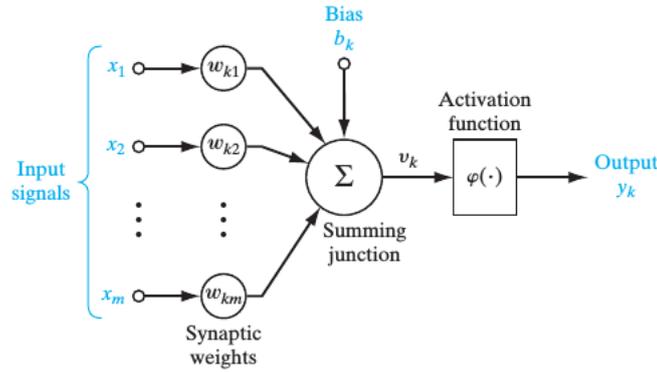


Figure 3.1 – Définition d’un neurone : unité de réinterprétation de l’information entrante permettant au réseau de la résumer et de la moduler dans un but plus global *d’apprentissage*. Les poids synaptiques, ainsi que le biais propre au neurone constituent ce que nous avons appelés les paramètres du réseaux (Haykin et al., 2009).

mentaux de cette théorie dont certains ont été écrits en italique dans le paragraphe précédent. Dans un second temps, nous parlerons des réseaux à convolutions et de leurs particularités.

Le cœur du réseau : le neurone - Formellement, un neurone est défini comme une unité de traitement du signal entrant dont le but est d’émettre un signal “activé” aux neurones de la couche suivante. Ce traitement du signal se fait en deux temps :

- Le neurone établit une combinaison linéaire des informations qui lui sont envoyées par les neurones d’une ou plusieurs couche(s) précédente(s), combinaison pondérée par des paramètres associés à chaque connexion entrante dans ce neurone. Dans l’analogie avec la neurobiologie, ces entrées issues d’unités précédentes et les poids associés constituent les synapses qui établissent des liaisons neuronales entre le neurone en question et d’autres neurones.
- Le neurone ajuste cette combinaison linéaire par l’intermédiaire d’un paramètre propre appelé biais, et réinterprète toute l’information reçue par une fonction non-linéaire dite d’activation. La fonction d’activation peut avoir plusieurs but selon sa forme, mais de manière générale elle empêche la saturation de l’ensemble du réseau, et facilite son apprentissage comme nous le verrons plus loin.

La figure Fig.(3.1) extraite de (Haykin et al., 2009) illustre ces deux étapes du process de l’information entrant par le neurone. Pour l’instant, considérons le fait que chaque couche de neurones est liée à sa voisine par une matrice de poids W_{ij}^l , l’exposant l signifie que ce sont les poids liant les couches $l - 1$ et l . La sortie émise par le $i^{\text{ème}}$ neurone de la $l^{\text{ème}}$ couche cachée s’écrit :

$$\sigma_i^l = \varphi_i^l \left(\sum_{j=0}^{M-1} W_{ij}^l \text{inp}_j^l + b_i^l \right) \quad (3.1.1)$$

Avec inp_j^l la $j^{\text{ème}}$ entrée de la $l^{\text{ème}}$ couche, on peut le voir comme la $j^{\text{ème}}$ sortie de la $l - 1^{\text{ème}}$ couche, b_i^l représente le biais propre au $i^{\text{ème}}$ neurone de la couche l .

Nécessairement, la matrice W_{ij}^l est de dimension $M \times N$ avec M le nombre de neurones de la couche l et N le nombre de neurones dans la couche d’entrée $l - 1$. Enfin, φ_i^l représente

la fonction d'activation du neurone considéré. C'est elle qui complexifie les relations entre chaque couche. Les fonctions comme la tangente hyperbolique (qui admet deux comportements asymptotiques) répartit la sortie de chaque neurone dans un intervalle de valeur $[-1; 1]$, elle est donc très utile quand il s'agit de classifier l'entrée dans une certaine catégorie. La fonction ReLU laissera exprimer le neurone qu'à partir du moment où la valeur de sa sortie est positive ; nous reviendrons sur les fonctions d'activations, leurs propriétés et leur impact plus tard (3.2).

La fonction de coût - Au terme de toutes les compositions des activations non linéaires, le signal d'entrée est transformé en un signal que l'on appelle la prédiction (c'est le signal de sortie du réseau). Lors du processus d'apprentissage, il est nécessaire d'évaluer la distance ou l'erreur entre la prédiction et la cible. Pour mesurer cet écart, nous utilisons une fonction de coût notée généralement \mathcal{J} . Le plus souvent, la fonction de coût choisie est la norme Euclidienne, parfois la valeur absolue. Dans les cas les plus complexes, il est nécessaire de combiner les distances pour obtenir une information plus riche sur l'origine des écarts entre la prédiction et la cible, permettant au réseau de mieux identifier les poids à ajuster pour améliorer la prédiction. On parlera de vision panoramique de la fonction de coût¹. Nous fournirons plus de détails section 3.3.

Apprentissage et rétro-propagation - L'apprentissage d'un réseau de neurone ou entraînement (training), est un processus itératif consistant en l'ajustement des poids et biais associés à chaque neurone et leurs inter-connexions dans le but de minimiser la fonction de coût sur l'ensemble des données d'apprentissage. C'est un processus qui est inspiré du fonctionnement du cerveau dont l'exemple le plus connu est l'apprentissage de la marche. L'ajustement des paramètres se fait par rapport à la sensibilité de la fonction de coût vis-à-vis de chaque paramètre du réseau, biais compris. Les gradients sont calculés durant la phase avant de l'apprentissage c'est-à-dire lorsque la prédiction sur une entrée s'effectue ; l'ajustement des poids se fait en propageant le signal d'erreur des couches les plus profondes aux couches les plus superficielles, c'est la phase rétro-propagative. Chaque itération, appelée époque, est donc composée de deux phases :

- *Phase avant ou d'excitation* : un ou plusieurs stimuli excite(nt) le réseau qui émet une réponse ; l'écart entre la réponse obtenue et celle attendue constitue l'erreur globale du réseau mesurée par \mathcal{J} . Les éventuelles métriques sont également évaluées ;
- *Phase rétropropagative de l'erreur ou backpropagation* : l'erreur mesurée est renvoyée de la sortie vers l'entrée, chaque gradient est calculé par composition des dérivées (chain rule). Il est alors possible de connaître la variation de la fonction de coût par rapport à chaque paramètre du réseau. Ces variations sont enfin utilisées pour mettre à jour les valeurs de ces paramètres. Cela clos une époque d'apprentissage.

Généralement nous considérons un apprentissage stochastique : un nombre N_{batch} de paires d'entrées et de sorties sont aléatoirement sélectionnées en une collection appelée batch, puis

1. La notion de landscape est souvent évoquée dans l'apprentissage de fonctionnelle complexe notamment dans les processus génératifs. Ici nous avons proposé la traduction vision panoramique renseignant bien le fait que combiner les distances permet de fournir au réseau une meilleure compréhension de ce qu'il doit minimiser/apprendre. On parlera d'alignement entre l'optimisation et l'objectif recherché.

fournies au réseau comme données d'apprentissage. Le réseau mettra à jour les poids selon l'échantillon qu'on lui propose. Le fait que cette collection soit de petite taille ajoute du bruit sur l'entraînement ce qui aura plusieurs conséquences, bénéfiques pour la plupart : robustesse de l'apprentissage, convergence plus rapide. Il est possible de perdre en consistance ; il est conseillé de tester les méthodes d'initialisation des poids ou bien de pré-entraîner les poids au cours d'un nombre limité d'époques, avec un taux d'apprentissage assez lent pour éviter une divergence du réseau. Une époque est alors accomplie lorsque la totalité des données disponibles ont été sélectionnées et fournies au réseau pour son apprentissage. Dans les faits si le cardinal de la base données est noté $\text{Card}(\mathcal{D})$, une époque correspond à $\text{Card}(\mathcal{D})//N_{\text{batch}}$ passes avant et arrière dans le réseau où $//$ désigne la division entière.

Si nous représentons les compositions d'activations par \mathcal{F} , les couches de neurones permettant de passer des signaux d'entrées X aux signaux de sorties Y , apprendre c'est trouver les paramètres optimaux θ^* minimisant la distance entre l'objectif et la solution prédite par le réseau :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{J}(Y, \mathcal{F}(X; \theta)) \quad (3.1.2)$$

Apprentissage supervisé et théorie Bayésienne - Il est intéressant de comprendre l'apprentissage d'après le formalisme Bayésien. Écrivons le théorème de Bayes pour le problème de l'apprentissage avec les notations déjà introduites en rajoutant les indices tr pour signifier qu'il s'agit des données d'apprentissage. Apprendre c'est maximiser la probabilité suivante :

$$p(\theta | X_{tr}, Y_{tr}) = \frac{p(Y_{tr} | X_{tr}, \theta) p(\theta)}{p(Y_{tr} | X_{tr})} \quad (3.1.3)$$

On définit la distribution de probabilité de vraisemblance (*likelihood*) de θ par

$$\mathcal{L}(\theta) = p(Y_{tr} | X_{tr}, \theta) \quad (3.1.4)$$

Elle représente la probabilité conditionnelle d'observer la cible Y_{tr} connaissant les entrées X_{tr} et les paramètres du modèle. La likelihood est pondérée par la distribution à partir de laquelle les paramètres du réseaux sont initialisés ; c'est l'information a priori $p(\theta)$. La probabilité $p(\theta | X_{tr}, Y_{tr})$, que l'on cherche à obtenir dans le processus de l'entraînement, est la distribution a posteriori des paramètres θ . Dans une logique d'apprentissage supervisée, cette probabilité est conditionnée par les entrées et les sorties fournies au réseau au cours de son apprentissage.² En vertu de l'équation Eq.(3.1.3), cette optimisation se fait au travers de la likelihood et l'initialisation des paramètres. Cette optique fournit les incertitudes et les moments statistiques sur les paramètres θ sachant la base de données proposée lors de l'apprentissage (Barber, 2012). Cette approche appelée Bayesian Learning est la base des processus génératifs (Goodfellow et al. (2014), Mirza and Osindero (2014)) et les approches variationnelles (Makhzani et al., 2015).

Inférence approximative et méthode de Maximum A Priori (MAP) - Le calcul exact de la probabilité a posteriori des paramètres θ est une tâche dont bien souvent le temps

2. Hors du cadre de la supervision, les distributions entières des variables sont requises. Voir par exemple Barber (2012).

d'exécution est exponentiel pour des réseaux profonds, mais pas uniquement (Goodfellow et al., 2016, chap. 19). Dans la majorité des cas (à l'exclusion donc, des méthodes variationnelles et génératives), il est suffisant de faire l'estimation des valeurs de θ^* telle qu'elle maximise le produit $\mathcal{L}(\theta)p(\theta)$. L'estimation n'est plus une distribution mais un point qui concentre la meilleure information de la fonction de vraisemblance selon l'initialisation des paramètres. Cela implique donc deux choses :

- Nous renonçons définitivement aux statistiques de la prédiction : moments statistiques, domaine d'incertitude ; ceci pour permettre la réalisation d'une tâche.
- L'initialisation des paramètres joue un rôle souvent déterminant dans l'obtention des résultats. Il existe d'ailleurs des articles décrivant les stratégies d'initialisation selon les cas. Les distributions les plus classiques sont décrites dans les articles suivants Glorot and Bengio (2010), Saxe et al. (2013) et He et al. (2015).

Cette estimation est une méthode bien connue et majoritairement utilisée hors problèmes génératifs et/ou variationnels, c'est la méthode MAP (maximum a posteriori). On écrit le problème d'apprentissage au sens de l'estimation ponctuelle :

$$\theta^* = \operatorname{argmax}_{\theta} \underbrace{p(Y_{tr} | X_{tr}, \theta)}_{\mathcal{L}(\theta)} p(\theta) \quad (3.1.5)$$

Pour l'estimation ponctuelle, les formulations (3.1.5) et (3.1.2) sont équivalentes.

Des techniques comme le batch normalisation (Ioffe and Szegedy, 2015) ou le transfert d'apprentissage (Zhuang et al., 2020) ont été développées en place pour contrer la forte sensibilité de l'apprentissage par rapport à l'initialisation des poids. La première consiste à réduire la variance des entrées de chaque couche pour chaque batch ; la deuxième initialise les paramètres du réseau à partir d'un pré-entraînement sur une tâche similaire. Au fur et à mesure que des chercheurs explorent l'initialisation, de plus en plus de "recettes" ou procédés ont été élaborés. Ces approches n'ont pas été fructueuses dans notre cas, ce qui arrive parfois (comme le rapporte le créateur de la batch normalisation dans une conférence, et plusieurs articles dont Simonyan and Zisserman (2014)). Nous avons utilisé la méthode de recherche aléatoire (Goodfellow et al., 2016) pour tester l'influence des méthodes d'initialisation sur l'évolution de la fonction de coût et des métriques sur un nombre limité d'itérations. Il ressort que pour certains algorithmes d'optimisation, l'initialisation dite orthogonal proposée par Saxe et al. (2013), encouragée par Goodfellow et al. (2016) dans les problèmes de régression, permet d'atteindre les meilleurs scores de convergence dans les meilleurs temps ; ce même procédé d'initialisation fait diverger l'optimisation pour un autre optimiseur. Ce n'est donc pas une science exacte et l'exploration des hyper-paramètres dont l'initialisation des poids est encore indispensable.

Un mot sur la généralisation - L'intelligence artificielle est fondamentalement tournée vers la notion de généralisation. Lorsque les paramètres d'un réseau sont optimisés et que la phase de learning se termine, nous devons répondre à une question fondamentale : est ce que le réseau à mémoriser ou est ce qu'il a appris. Pour cela, il faut évaluer la qualité de la prédiction effectuée lorsque des entrées inédites lui sont proposées. Par inédites, on entend généralement qu'elles soient différentes de la base de données fournie lors de l'apprentissage ; différentes mais reposant malgré tout sur les mêmes principes. À partir de cette évaluation, deux scénarios sont possibles :

- Lorsque l'apprentissage est très performant mais que le réseau est incapable de prédire des solutions satisfaisantes sur l'espace de test : c'est la mémorisation ou sur-apprentissage (*overfitting*)
- Lorsque le réseau est également performant sur l'apprentissage et qu'il arrive à l'être sur la dataset de test. On parlera alors de généralisation.

Bien souvent, la mémorisation est la preuve que le modèle utilisé pour apprendre la correspondance entre entrées et sorties est plus complexe que nécessaire. Le réseau utilise l'excédent de complexité (ou degrés de liberté) pour apprendre les "bruits" des données c'est-à-dire tous les comportements atypiques propre à la base de données utilisée pour l'apprentissage, on dit que le modèle a sur-appris. Paradoxalement, le modèle au préalable trop complexe, devient propre aux données utilisées empêchant son application aux autres bases de données. La mémorisation est évidemment indésirable.

Lorsque le réseau présente des capacités de généralisation, il présentera également des scores d'apprentissage plus bas que le réseau qui aurait sur-appris. D'après les remarques précédentes, nous pouvons interpréter cette moindre performance par le fait que le réseau a du sélectionner les informations les plus capitales pour comprendre la relation entre les entrées et les sorties dans son ensemble, quitte à ne pas considérer les comportements les plus rares. C'est le processus itératif qui force³ cette perte d'information. Ce raisonnement peut éventuellement pousser à l'inverse de l'overfit qui est appelé l'*underfit* ou sous-apprentissage : cela arrive lorsque le réseau n'est pas assez complexe pour découvrir ou conserver suffisamment d'information mutuelle entre entrée et sortie. Nous reviendrons sur la notion de généralisation dans la section 3.3.

3.1.2 Les réseaux de neurones à convolutions

Les réseaux de neurones à convolutions, ou CNN pour Convolutional Neural Network, constituent une classe particulière de réseau de neurones. En effet, on retrouve les couches de neurones liées entre elles par des poids, appelés filtres, chaque représentation intermédiaire ainsi créée est activée et ainsi de suite jusqu'à ce que le réseau émette une prédiction. Leur entraînement se fait traditionnellement également par rétropropagation. Seulement, entre 2010 et 2012, les travaux de [Krizhevsky and Hinton \(2010\)](#), [Mohamed et al. \(2011\)](#), [Dahl et al. \(2011\)](#) et enfin [Krizhevsky et al. \(2012\)](#) mirent en évidence que les CNN surpassent les réseaux de neurones classiques lorsque la prédiction nécessite un contexte comme la reconnaissance vocale - faciale etc. À partir de 2012, la popularité des CNN explosa. Aujourd'hui avec les progrès toujours plus grands et rapides des capacités de calcul, ils sont utilisés dans presque tous les domaines de la reconnaissance vocal à la génération de peinture en passant par la reconnaissance faciale, les voitures autonomes et très récemment la physique. Ce sont d'ailleurs ce type de réseau de neurones que nous considérerons. Dans la suite de ce manuscrit nous désignerons les réseaux de neurones à connexions matricielles précédemment présentés par l'abréviation ANN pour Artificial Neural Network. Les paragraphes suivants sont inspirés entre autres de ([Goodfellow et al., 2016](#), chap. 9).

3. C'est un procédé non supervisé qui est identifié comme le meilleur moyen de minimiser l'erreur commise sur la prédiction.

Propriétés du CNN - Contrairement aux ANN, les CNN admettent en entrée des données structurées sous forme de vecteurs, de matrices ou de volumes selon le type de CNN (1D, 2D ou 3D respectivement) ; la convolution préserve également cette structure spatiale.

Ils sont composés de couche de convolutions : des filtres sont convolués avec l'entrée de la couche, on dira que le filtre glisse spatialement sur l'entrée. À chaque position du filtre, un neurone de la couche de convolution effectuera les multiplications et additions conformément à la formule de la convolution rappelée équation Eq.(3.1.6). Comme pour les ANN, chaque neurone "activera" son signal de sortie. La représentation de l'ensemble de neurones est appelée carte d'activation *activation map* ou carte des caractéristiques *feature map*.

À partir de cette opération de convolution, nous pouvons identifier deux propriétés inédites aux CNN :

- La connectivité sporadique ou *sparse connectivity* : chaque neurone n'aura qu'une portion limitée des entrées conformément aux dimensions du filtre ; on parlera alors de connectivité sporadique à condition que les filtres soient plus petits que les entrées. Cette propriété permet par exemple de trouver des détails de la taille d'une dizaine ou centaine de pixels dans une image de plusieurs millions de pixels. On peut tout de suite imaginer qu'un CNN est théoriquement capable d'identifier les structures turbulentes aux différentes échelles et d'en prédire les conséquences au niveau énergétique ou autre.
- La partage des paramètres ou *weight sharing* : les valeurs qui constituent les filtres sont les paramètres du réseaux de neurones ; ce sont ces valeurs qui sont ajustables. Chacune des valeurs du filtre sera utilisée sur la totalité des points de l'entrée. Ainsi, au lieu d'apprendre les valeurs de paramètres propres à chaque position spatiale (comme cela est nécessaire pour les ANN), un seul ensemble de valeur est appris. Le gain de temps s'effectue surtout sur la rétropropagation puisque le nombre de paramètres à ajuster est fortement réduit.

Au plus le nombre de filtres de la couche de convolutions est élevé au plus le nombre d'activation maps obtenues sera élevé. La figure Fig.(3.2) représente l'opération de convolution effectuée sur une image exprimée en couleur RGB. Notons que le filtre traverse également les composantes des entrées. La représentation résultante prendra donc en compte les détails de l'ensemble de la donnée d'entrée.

Dans un cas 2D, nous définissons la formule de convolution par

$$o^l(i, j) = (\text{inp}^l * K^l)(i, j) = \varphi^l \left(\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \text{inp}^l(i+m, j+n) K^l(m, n) + b^l \right) \quad (3.1.6)$$

avec K le filtre (ou kernel) de dimensions $M \times N$, l'opérateur $*$ symbolise la convolution et l'exposant l symbolise le numéro de la couche de convolution auquel le neurone appartient, φ et b représentent respectivement la fonction d'activation et le biais associés au neurone. Enfin i et j sont des coordonnées localisant le neurone.

Une notion clé : le champ réceptif - La propriété de connexion sporadique est soulignée par l'équation Eq.(3.1.6) : le filtre se concentre sur une zone centrée autour de la position (i, j) , en occultant les autres parties de l'entrée. La zone sur laquelle le neurone calcule la convolution est appelée champ réceptif. Lorsque deux couches de convolutions ou plus sont mises l'une à

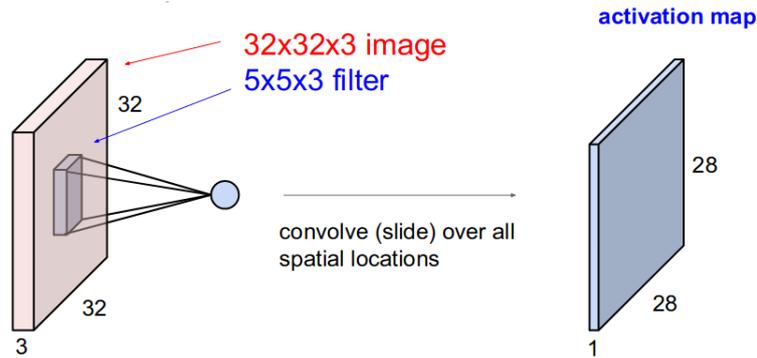


Figure 3.2 – Représentation de la convolution d’un filtre de taille 5x5 avec une entrée composée de trois matrices de taille 32x32. La carte d’activation est formée à partir de cette convolution sur toute l’entrée. On note que le filtre possède une dimension supplémentaire égale au nombre de matrices en entrée ; la convolution se fait simultanément sur toutes les composantes de l’entrée. Image issue de http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture05.pdf.

la suite de l’autre, les neurones de la deuxième couche ont un plus grand champ réceptif que ceux de la première, puisqu’ils convoluent la convolution de l’entrée par la première couche. Nous pouvons déduire alors que les patterns des couches plus profondes sont le résultat de combinaisons des structures identifiées par les couches de convolutions plus superficielles. Les patterns sont plus complexes ; on parlera de hiérarchisation non supervisée des patterns (Liu et al., 2017). Ainsi, en principe, plus on empilera de couches de convolutions, au plus le champ réceptif des neurones sera grand et au plus la prédiction (ou reconstruction) du neurone sera fiable puisque plus raffinée. Pour l’identification du pattern, le réseau fait “glisser” un filtre d’une taille fixe sur l’entrée de chaque couche de neurone. Les valeurs du filtres sont ici les paramètres du réseau. Chaque neurone effectue une convolution du filtre avec les valeurs de l’entrée que le filtre recouvre ; au fur et a mesure que le filtre traverse le domaine spatial des entrées, les neurones de la couche cachée émettent une sortie.

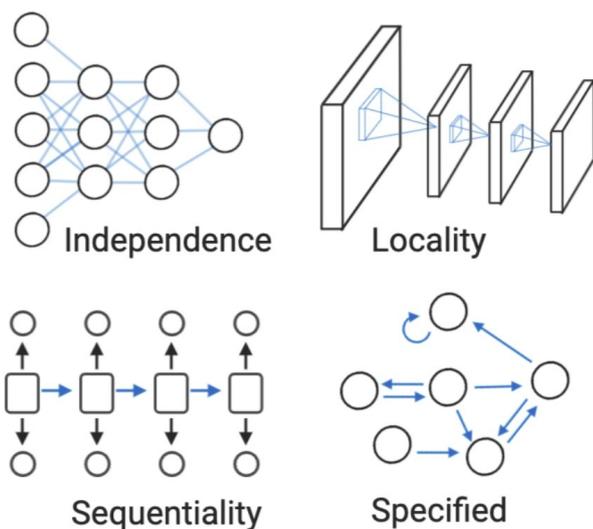


Figure 3.3 – Propagation de l’information selon les connexions entre couches (Finlayson, 2020). Les ANN considèrent que les entrées de chaque couche sont indépendantes des entrées des couches précédentes. Au contraire le CNN construit des couches de patterns plus complexes les unes que les autres et conservent une forme de mémoire des entrées des couches depuis l’entrée globale. Les Recurrent Neural Networks (RNN) (figures du bas) cherchent des liens dynamiques entre les époques d’entraînement en faisant circuler les informations des époques précédentes. Cela peut se faire séquentiellement ou de manière plus spécifiée.

Composition des dérivées et problème de d’extinction du gradient - De manière générale, qu’il s’agisse de ANN ou de CNN (et encore plus quand il s’agit de réseaux de neurones récurrents qu’on n’abordera pas dans ce manuscrit ; voir figure Fig.(3.3)), il faut impérativement éviter les problèmes de gradients nuls ou presque, en cours d’apprentissage. Ce problème que nous appellerons *vanishing gradient* provient généralement d’un manque de précaution par rapport à la formule des dérivées composées ou *chain rule*. En effet, la formule de la dérivée des fonctions composées permet d’écrire les gradients de la fonction de coût comme le produit des dérivées de \mathcal{J} par rapport aux paramètres du réseaux. Au fur et à mesure que le signal d’erreur arrive vers les couches superficielles, les gradients sont déjà la résultante de plusieurs milliers de produits, il est donc très probable que le signal soit trop faible pour modifier significativement et pertinemment les poids de ces couches superficielles gelant une partie du réseau puis petit à petit tout le réseau. Plusieurs remèdes ont été proposés, notamment la normalisation des batch (ou patch) (Ioffe and Szegedy, 2015) de données, empêchant ce genre de problèmes en reparamétrant l’entrée de chaque couche pour qu’elle soit de moyenne nulle et de déviation unitaire. Cela réduit la dépendance des gradients entre différentes couches permettant au signal de correction de mieux se propager. Notons également que la normalisation des batch introduit du bruit dans le réseau qui au final le rend plus robuste. D’autres méthodes telles que le Dropout (Gal and Ghahramani, 2016) sont utilisées forçant le réseau à construire des patterns robustes et réduisant l’extinction de gradient. Une autre façon de gérer ce dernier problème est d’établir une stratégie de corrections des paramètres du réseau. Les méthodes jouent sur le calcul du gradient, méthodes d’optimisation dont le taux d’apprentissage est adaptée selon chaque poids ; nous en parlerons plus tard section 3.3.3.

Compression et décompression par convolutions - Une autre opération propre aux CNN est la couche de pooling. Le pooling consiste à réduire la dimension spatiale de l’information se propageant dans le réseau. Par exemple, si nous considérons un cas 2D d’une entrée de taille (16×16) , l’opération de pooling effectuée sur une fenêtre de taille (2×2) renverra une représentation de taille 8×8 . Le plus couramment dans la littérature l’opération la plus courante pour cette réduction est le Max-Pooling (Scherer et al., 2010). En reprenant notre exemple, le Max-Pooling renverra une représentation 8×8 constituée du maximum détecté dans chacune des fenêtres considérées.⁴ Selon (Goodfellow et al., 2016, chap.9), les couches de poolings peuvent être utilisées pour induire des invariances par rotations dans le réseau, mais certaines précautions doivent être prises. De façon plus directe, un réseau qui est entièrement à convolutions est invariant aux légères translations des entrées (bruit uniforme). Il s’avère que cette propriété peut se perdre si des opérations qui n’impliquent pas des convolutions sont utilisées et en particulier le pooling. Même si cette perte de robustesse n’est pas systématique, nous avons opté pour un pooling généralisé à l’aide de convolutions à pas ou *stride convolutions* dont le pas dans l’espace des entrées $s = 2$.⁵ Lorsque l’on utilise les convolutions, il faut avoir en tête la formule reliant la taille de la sortie N_O aux tailles de l’entrée N_I , du filtre N_F et du

4. L’Average-Pooling a été longtemps utilisé. Notons qu’il existe aussi des opérations plus complexes comme le Max-Pooling Généralisé (Murray and Perronnin, 2014). D’autres (Santos et al., 2016) proposent de sélectionner la grandeur la plus importante relativement à un score.

5. D’après (Zhang, 2019), les convolutions à pas n’assurent pas systématiquement l’invariance par translation.

pas s de la convolution par la formule :

$$N_O = \frac{N_I - N_F}{s} + 1 \quad (3.1.7)$$

Sans précaution, le code rognera automatiquement l'entrée voire affichera un message d'erreur. Une façon de contourner ce problème est le padding de zéros. L'option "same" disponible sur Keras ([Chollet et al., 2015](#)) assure de s'affranchir de tout type de problème⁶. La technique du padding permet d'utiliser pleinement la puissance de la convolution à pas : le réseau décidera de façon non-supervisée quels critères considérés pour une compression optimale. Évidemment cela se fera au prix de paramètres supplémentaires à optimiser. De façon analogue, la partie décompression des données se fait par interpolation bilinéaire (c'est l'opération d'up-sampling). Nous avons également préféré l'opération de convolution transposée avec un pas de 2. Cette opération est l'analogue de la convolution, elle permet alors au réseau de définir quelle est la meilleure stratégie d'Upsampling. Pour plus d'informations sur la convolution transposée, la convolution diluée et toutes les combinaisons possibles voir [Dumoulin and Visin \(2018\)](#).

Nous avons introduit les concepts fondamentaux des réseaux de neurones à convolutions. Si il fallait donner une limite aux réseaux de neurones, nous citerions le manque de règles systématiques pour la construction de réseau. Le grand nombre d'hyper-paramètres est donc systématiquement à explorer lors de la conception d'un modèle. Au cours de la dernière décennies de nombreuses études ont été dédiées à l'impact d'un ou plusieurs hyper-paramètres sur les capacités d'apprentissage et de généralisation. L'exploration des hyper-paramètres a constitué une grande partie de nos travaux de thèse puisque les méthodes présentes dans la littérature de la Turbulence-AI n'étaient pas probantes par rapport à notre projet. Nous avons alors étendu nos recherches aux articles d'intelligence artificielle pure comme ceux discutant de nouvelles fonctions d'activation ou d'optimiseurs prometteurs, en plus des architectures générales des CNN. Nous voulons détailler les propriétés générales que nous avons pu recueillir permettant d'expliquer pourquoi un ensemble d'hyper-paramètres pourrait permettre une meilleure convergence qu'un autre. Une partie de ces tests sera présenté dans le chapitre 4.

Dans les sections suivantes, nous présentons les fonctions d'activation en se focalisant sur leurs propriétés et leurs impacts sur la propagation du signal. Nous introduisons à la communauté de la Mécanique des Fluides les fonctions d'activation Swish et Mish qui dépassent de loin les fonctions d'activations les plus classiques.

3.2 Activations saturées ou sporadiques : fonctions d'activations

Les fonctions d'activation sont en général juxtaposées à la couche de neurones. On parle d'activation car elle va activer ou éteindre des neurones selon les signaux de sortie émis par ceux ci. Selon la forme de la fonction d'activation, nous pouvons déduire l'impact de ses propriétés sur la propagation de l'information au sein du réseau et l'objectif final.

6. Le padding s'incorpore dans la grandeur N_I comme $N_I = N_I + 2N_p$ où N_p est le nombre de ligne, colonne et profondeur de zéros à rajouter. Ceci est vrai dans le cas symétrique (d'où le facteur 2) qui nous intéresse. Plus généralement, on résoudra l'équation (3.1.7) pour chaque dimension spatiale : $(N_I)_{x,y,z}$.

Tangente hyperbolique et sigmoïde et leurs propriétés - Nous écrivons les formules et traçons leur forme respective (Fig.(3.4)) :

$$S(x) = \frac{1}{1 + e^{-x}} \quad (3.2.1)$$

$$T(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.2.2)$$

Ces deux fonctions sont dites à double saturations car elles admettent deux valeurs asymptotiques réelles. La fonction sigmoïde impose plus de zéros impliquant une creusalité (ou *sparsity*) des activations. Intuitivement cette façon d'activer les représentations tend à clusteriser les données ; d'ailleurs ces fonctions d'activations sont souvent utilisées dans les domaines de classification.

Les fonctions à double saturation nécessitent d'être précautionneux lors de l'initialisation des poids notamment pour que leurs entrées restent dans le régime linéaire (Ramachandran et al., 2017). Une façon d'optimiser leur utilisation est de les combiner avec une renormalisation des entrées avant activation comme le batch normalisation Ioffe and Szegedy (2015), ceci aura pour effet supplémentaire de réduire l'impact de l'initialisation des poids sur l'apprentissage.

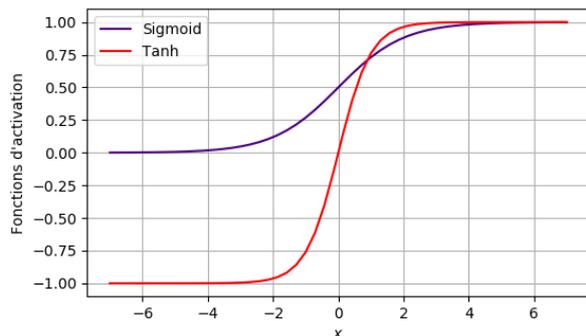


Figure 3.4 – Ci contre - tracés des fonctions tangente hyperbolique et sigmoïde.

Sans ces précautions, les queues de ces fonctions ont tendance à éteindre les gradients à cause d'une possible saturation du réseau. Ce qui est très indésirable.

Unité rectificatrice linéaire ReLU - Les unités rectificatrices linéaires ou ReLU sont les fonctions d'activation les plus largement utilisées dans toutes les catégories d'apprentissage. Introduites par (Hahnloser et al., 2000) se basant sur l'activité cérébrale pour la reconnaissance de pattern et d'une analyse, elles sont bien souvent présentes dans les modèles les plus performants depuis ImageNet (Krizhevsky et al., 2012). La fonction ReLU est représentée par la courbe bleue figure Fig.(3.5). On l'écrit :

$$\text{ReLU}(x) = \begin{cases} x & x \geq 0, \\ 0 & \text{Sinon.} \end{cases} \quad (3.2.3)$$

Dans sa version originale, la ReLU possède beaucoup de propriétés intéressantes : tout d'abord contrairement aux deux fonctions précédentes, la ReLU n'est saturante qu'à gauche de zéro.

Toutes les activations négatives sont oubliées induisant une pénalisation sur l'apprentissage favorisant la généralisation. Elle augmente la creusité des activations et permet à l'information de mieux se propager dans le réseau que les fonctions sigmoïde et tangente hyperbolique, et réduit les risques d'extinction ou d'explosion de gradient. Cependant, les activations produites par cette fonction sont souvent fortes et il est recommandé ici aussi d'utiliser des renormalisation des entrées après chaque activation. Au fur et à mesure de leur utilisation, le problème majeur des ReLU appelé *dying ReLU problem* a été identifié. Les réseaux entraînés avec la fonction ReLU peuvent devenir inactifs et la sortie du réseau sera alors intégralement zéro. Il semblerait que l'origine de ce problème soit les forts gradients qui peuvent donner une valeur de biais très inférieure à zéro, pour certains neurones (souvent près de 40% des neurones si aucune précaution n'est prise). Comme on l'a vu, le biais ajuste les informations arrivant dans le neurone avant que ce dernier n'émette son signal de sortie activé. Ce biais, très négatif translaterait alors toute entrée dans le régime d'inactivité du neurone. Au fur et à mesure les neurones s'éteignent et le réseau est complètement bloqué et impossible de discerner une entrée d'une autre. Les renormalisations peuvent aider à contourner ce problème, mais elles ajoutent également de la complexité à l'apprentissage en plus de paramètres entraînaibles supplémentaires. Elles dégradent parfois l'apprentissage.

Ajustement : quelques variantes du ReLU - Les chercheurs ont essayé de contourner le problème des extinctions de réseau en créant des variantes plus ou moins complexes du ReLU classique. Une des variantes les plus efficaces et ne rajoutant aucun paramètres ajustables est la fonction Leaky-ReLU (Maas et al., 2013), elle dépend d'un hyper-paramètre non ajustable

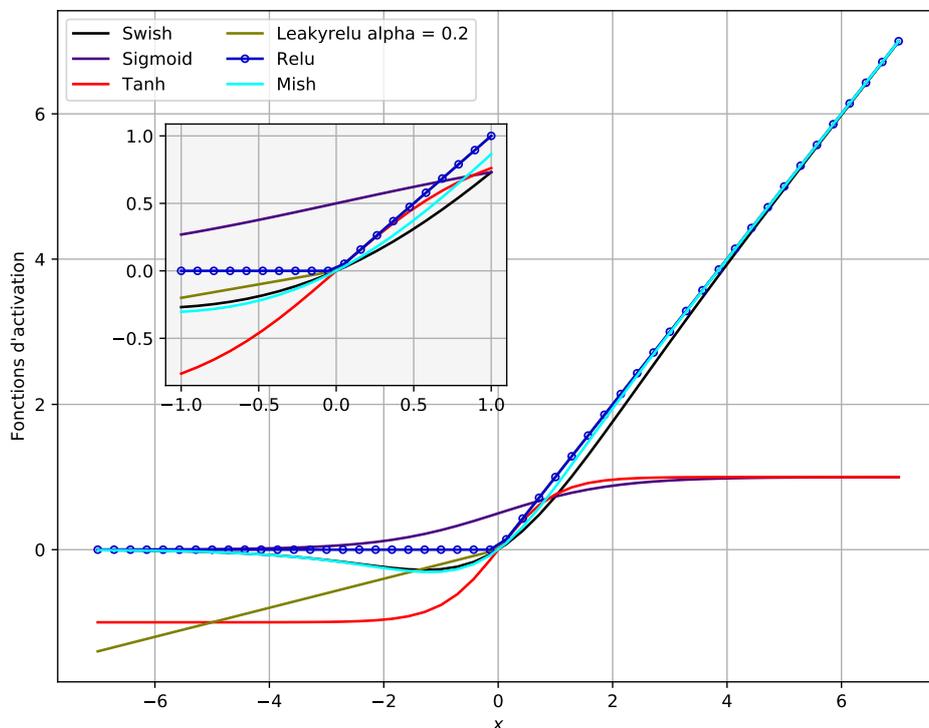


Figure 3.5 – Graphiques de différentes fonctions d'activation, des plus classiques comme la Sigmoïde (ligne violette) jusqu'aux plus exotiques comme l'activation Mish (ligne cyan).

α et s'écrit

$$\text{Leaky-ReLU}(x) = \begin{cases} x & x \geq 0, \\ -\alpha x & \text{Sinon.} \end{cases} \quad (3.2.4)$$

La valeur recommandée pour α est 0.2 ; on trace cette fonction avec cet alpha Fig.(3.5) courbe kaki. Le fait d'activer des signaux négatifs va tirer la moyenne des activations de la couche de neurones vers 0 accélérant au passage l'apprentissage (Maas et al., 2013). Forcément, puisque la fonction d'activation n'est pas saturante, le problème du dying ReLU n'existe plus. Même si nous conservons une pente α plutôt petites, nous perdons quand même en creusalité. De plus cette pente, si elle est mal initialisée peut entraîner un apprentissage sous-optimal, c'est l'inconvénient qui a rendu cette activation peu populaire (Liu et al., 2019).

La variante du ReLU qui est souvent utilisée est une ReLU paramétrique, on l'appelle PReLU (He et al., 2015). Comme son nom l'indique, la PReLU propose d'ajuster la pente de chaque unité rectificatrice de façon non supervisée. Le prix à payer est l'augmentation du nombre de paramètres ajustables. On l'écrit :

$$\text{PReLU}(x) = \begin{cases} x & x \geq 0, \\ -\alpha_i x & \text{Sinon.} \end{cases} \quad (3.2.5)$$

avec α_i étant la pente pour le $i^{\text{ème}}$ neurone d'une couche de neurones. Le PReLU entend donc résoudre les inconvénients de la ReLU originale tout en évitant les problèmes d'initialisation des pentes de la fonction Leaky-ReLU.

Il existe d'autres unités rectificatrices citons par exemple les unités exponentielles (Clevert et al., 2015), les unités auto-normalisantes (SeLU Klambauer et al. (2017)). Ces deux dernières sont très utiles dans les réseaux à connexions denses, mais elles se sont avérées inefficaces dans les travaux que nous présentons ici.

Activations composées Swish et Mish - La littérature regorge d'idées innovantes et notamment de fonctions d'activations aux propriétés toujours plus intéressantes. Elles sont en générales composées de fonctions de classiques et utilisent les forces des unes pour compenser les faiblesses des autres. La fonction Swish introduite par Ramachandran et al. (2017) s'écrit :

$$\text{Sw}(x; \beta) = \frac{x}{1 + e^{-\beta x}} = x S(\beta x) \quad (3.2.6)$$

Avec $S(x)$ la fonction sigmoïde définie équation Eq.(3.2.1). Son évolution en fonction de l'entrée x est illustrée par la courbe noire Fig.(3.5) (avec $\beta = 1$ fixé). Cette fonction se comporte comme la ReLU pour les entrées positives, dans le sens où elle n'est pas bornée. Pour les entrées suffisamment négatives, cette fonction se comporte comme la sigmoïde augmentant la creusalité de l'activation. Enfin dans une zone qui est régie majoritairement par le paramètre β , la fonction Swish permet l'expression des entrées plus ou moins négatives permettant d'éviter de tirer vers zéro la moyenne des activations de la couche de neurones. Ces valeurs négatives permettent une plus grande expressivité du réseau. Une propriété perdue par les unités rectificatrices est la continuité en tout point du domaine de définition permettant au gradient de mieux circuler. Plus d'informations sur l'impact de ces propriétés dans les premières itérations de l'apprentis-

sage sont disponibles dans le papier originale [Ramachandran et al. \(2017\)](#)⁷. Dans nos travaux, le β est fixé à 1, mais il s'est avéré que ce n'est pas toujours la valeur la plus optimale. Malgré tout, elle permet une meilleure convergence du réseau de neurone qu'avec toutes les fonctions d'activations précédemment présentées.

La fonction d'activation Mish ([Misra, 2019](#)) s'est avérée être encore plus optimale pour nos travaux que la fonction Swish. On l'écrit :

$$M(x) = x T(\ln(1 + e^x)) \quad (3.2.7)$$

Elle est représentée par la courbe cyan Fig.(3.5). Son comportement est très proche de la fonction Swish. Elle est également non bornée à droite. Elle se confond d'ailleurs plus rapidement au ReLU que la Swish. Elle est également continue, non monotone et bornée par zéro dès lors que l'entrée est trop négative. Nous ignorons fondamentalement pourquoi un réseau utilisant la fonction Mish propose des résultats presque systématiquement meilleurs que celui utilisant la fonction Swish, puisqu'elles sont similaires. Les auteurs du papier présentant cette l'activation montrent que les dérivées premières et secondes des activations Swish et Mish sont assez similaires ; cependant la fonction Mish a tendance à être plus lisse dans la représentation des entrées empêchant toute singularité ou discontinuité pouvant éventuellement gêner la propagation des gradients, qui plus est en début d'apprentissage.

3.3 Objectif et alignement : fonction de coût, régularisation et méthodes d'optimisation

3.3.1 La fonction de coût et les métriques

Nous avons défini la fonction de coût comme l'outil de mesure de l'erreur instantanée commise par le réseau. À partir de cette évaluation, le signal d'erreur se propagera et par dérivées composées, les sensibilités de la fonction de coût par rapport aux paramètres du modèle sont calculées. À partir de ces calculs, les algorithmes d'optimisation mettent à jour, c'est-à-dire corrigent, les valeurs de ces paramètres. On comprend que si cette mesure d'erreur ne fournit pas assez d'information sur l'état de l'apprentissage, il est peu probable que le réseau converge. En général, pour des problèmes de classification les choix sont restreints et gravitent autour du score de précision de la prédiction. En régression, il faut être plus prudent : dans le cas de prédictions de faibles amplitudes, la norme L_1 de l'erreur est plutôt recommandée, alors que la norme L_2 semble adaptée pour les autres cas. Ces deux mesures de l'erreur s'écrivent :

$$L_1 = \sum_{\mathcal{D}} |Y_{\text{True}} - Y_{\text{Pred}}(\theta)| = \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_1 \quad (3.3.1)$$

$$L_2 = \sum_{\mathcal{D}} (Y_{\text{True}} - Y_{\text{Pred}}(\theta))^2 = \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_2 \quad (3.3.2)$$

7. [Liu et al. \(2019\)](#) expliquent que la sigmoïde aide à la mémorisation des patterns construits dans les couches de convolutions précédentes et lient cette propriété aux compositions des activations que l'on retrouve dans les réseaux récurrents appelés LSTM ([Staudemeyer and Morris, 2019](#)).

Cette précaution est due au fait que le carré de faibles amplitudes sera d'autant plus faibles, et la composition des dérivées exprimant la dérivée de la fonction de coût par rapport à un paramètre comme le produit d'un certain nombre de dérivées en partant de \mathcal{J} , il est très probable que le signal tende rapidement vers zéro rendant inefficent l'apprentissage. Dans le cas où l'apprentissage semble converger mais que des erreurs persistent, il est parfois nécessaire d'avoir recours à des métriques pour tester l'apprentissage sous différents éclairages. Imaginons une tâche de régression dont la fonction de coût est l'écart au carré entre prédiction et cible (abrégé L_2 ou MSE pour Mean Squared Error ou moindres carrés). Après n itérations, le score sur les moindres carrés est $2.67 \cdot 10^{-4}$; une métrique mesure de l'écart en valeur absolue (MAE pour mean absolute error) : $9 \cdot 10^{-3}$. Pour la même configuration, nous utilisons cette fois la MAE en guise de fonction de coût et la MSE comme métrique. À la fin des n itérations, nous mesurons l'écart en valeur absolue $2 \cdot 10^{-3}$ et sur la métrique $8.46 \cdot 10^{-6}$. On conclut qu'en optimisant le réseau avec la MAE, on arrive à une meilleure minimisation de cette erreur, y compris selon la MSE. Rappelons que l'évolution de la métrique n'intervient jamais dans l'apprentissage. Cet exemple illustre non seulement l'impact du choix de la fonction de coût, mais également l'utilité des métriques dont les scores apparaissent comme complémentaire à ceux de la fonction de coût. Cette illustration est issue d'un cas réel qui figure dans le tableau Tab.(4.4). En se basant sur Liu et al. (2020), nous avons également considéré des fonctions de coût plus complexes dites *customisées*; l'idée est de combiner les mesures de distances (norme L_1 , L_2 etc.) pour obtenir une optimisation des poids tenant compte de nuances et de comportements composés.

Fonctions de coût customisées aux problèmes considérés - Plutôt que de considérer toute l'information de différentes fonctions, nous avons tenté de pénaliser les écarts maximum présent dans une des deux métriques, et d'additionner ces écarts à la MAE déjà performantes. Ainsi, nous forçons le réseau à aller encore plus loin dans l'espace des paramètres et de converger vers des solutions, qui en plus de minimiser la norme L_1 , minimise les maxima rencontrés sur l'une des deux normes. Les meilleurs résultats ont été obtenus avec la fonction de coût suivantes appelées L_1L_{2M} :

$$L_1L_{2M} = \Omega \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_1 + (1 - \Omega) \max \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_2 \quad (3.3.3)$$

Cette moyenne est pondérée par l'hyper-paramètre Ω qui détermine l'impact de la pénalisation du maximum de l'erreur L_2 mesuré sur l'ensemble des points du patch de données. En reprenant notre cas de régression, l'erreur après n itérations sur la fonction de coût L_1L_{2M} est $2.2 \cdot 10^{-3}$, sur la L_1 la métrique donne $1.3 \cdot 10^{-3}$ et sur la L_2 : $7.66 \cdot 10^{-6}$. L'amélioration sur la norme L_1 est de près de 70 % simplement en rajoutant l'information sur les maxima de la norme L_2 . L'impact au niveau du temps de calcul ne se fait quasiment pas ressentir. On dira que l'optimisation du réseau est dans l'alignement de ce que l'opérateur (humain) attend de cette tâche. Une fonction de coût L_{1M} analogue pénalisant cette fois le maximum des écarts des valeurs absolues permet des performances similaires à la fonction L_1L_{2M} . Elle s'écrit

$$L_{1M} = \Omega \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_1 + (1 - \Omega) \max \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_1 \quad (3.3.4)$$

Des articles comme Pan and Duraisamy (2018) utilisent des pénalisation sur la Jacobienne de la fonction de coût (initialement MSE) par rapport aux entrées du réseau. D'après les auteurs, ce rajout induit une meilleure invariances aux légères translations des entrées réduisant

l'accumulation et la diffusion des erreurs inhérentes à chaque prédiction. Cependant, un mauvais paramétrage du facteur d'impact de cette pénalisation peut uniformiser la réponse du réseau. Coupler plusieurs façons d'estimer l'erreur commise par la prédiction est donc une bonne piste à suivre pour obtenir des réseaux plus performants voire induire des invariances à la fonctionnelle résultante. Cependant, chaque contrainte supplémentaire ajoute un hyper-paramètre et des risques de sous-optimalité deviennent de plus en plus probables. Dans les pires cas, la contrainte supplémentaire l'emporte sur la fonction objective et bloque l'apprentissage. Il faut donc être précautionneux et ne pas hésiter à tester des plages de valeurs différentes pour les hyper-paramètres.

3.3.2 Régularisation sur les paramètres du réseau

En théorie de l'optimisation, nous pouvons définir deux types d'optimisation : libre ou contrainte. Tels qu'ils ont été écrits, les algorithmes d'optimisation des ANN (ou CNN) ne tiennent pas compte de contraintes supplémentaires. On pourrait rajouter des étapes pour rendre conditionnelle la minimisation ; mais nous pouvons également rajouter des contraintes sur l'optimisation en utilisant l'approche de Karush-Kuhn-Tucker généralisant celle des multiplicateurs de Lagrange. Elle permet de prendre en compte les contraintes imposées par le système étudié et d'influencer la prise de décision conformément à l'objectif et aux contraintes. C'est l'idée de la régularisation sur une fonction de coût, pour une introduction plus complète voir [Goodfellow et al. \(2016, chap. 7\)](#). Toute la discussion précédente sur les pénalisation des fonctions de coûts peut également se comprendre par le formalisme des multiplicateurs de Lagrange. De manière générale, la régularisation consiste à effectuer un compromis entre erreur d'apprentissage et capacité de généralisation, ce que le réseau de neurones n'est pas encouragé à faire sinon. Ceci est d'autant plus vrai lorsqu'il s'agit de la régularisation sur les paramètres du réseau. Dans la pratique, il est préférable un réseau régularisé sur ses poids même trop profond, qu'un réseau de taille optimale sans régularisation sur les poids.

Impact des régularisations Lasso (L_1) et Ridge (L_2) sur la solution de l'apprentissage

- La régularisation sur les paramètres peut prendre plusieurs formes selon les objectifs. Généralement les poids du réseau sont contraints de sorte que ce dernier "éteigne" les connexions (poids) les moins actifs ou qu'il force un travail de groupe pour la reconnaissance de patterns. Nous distinguons deux types de pénalisation sur les poids :

- La pénalisation L_1 appelée *Lasso* pour Least Absolute Shrinkage and Selection Operator. Si \mathcal{J} est la fonction que l'on cherche à minimiser (customisé ou non), l'optimisation sous contraintes Lasso s'écrit :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} [\mathcal{J} + \lambda_{L_1} \|\theta\|_{L_1}] \quad (3.3.5)$$

avec λ_{L_1} est le coefficient d'impact de la régularisation Lasso.

- La pénalisation L_2 appelée *Ridge*. Cette fois l'optimisation sous contrainte Ridge de la fonction de coût s'écrit

$$\theta^* = \underset{\theta}{\operatorname{argmin}} [\mathcal{J} + \lambda_{L_2} \|\theta\|_{L_2}] \quad (3.3.6)$$

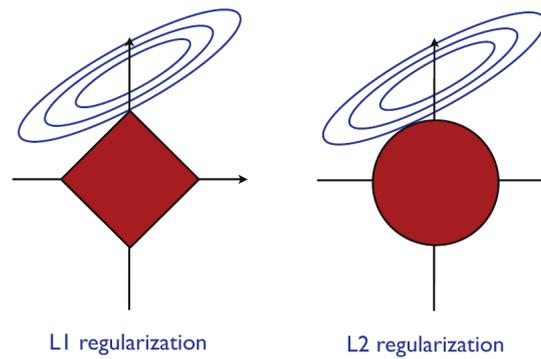


Figure 3.6 – Visualisation des normes L_1 et L_2 dans la recherche des poids optimaux (Mohri et al., 2018).

Ici aussi le facteur λ_{L_2} est un réel mesurant l'impact de la minimisation⁸.

Les impacts de l'une ou l'autre peuvent être visualisés Fig.(3.6) (Mohri et al., 2018). Dans cette figure, la fonction de coût est une MSE dont les contours sont représentés par les ellipsoïdes bleues. Les aires rouges délimitent les sphères au sens topologique des normes L_1 et L_2 respectivement. Le tout est tracé en fonction de deux paramètres pour illustrer l'idée générale de la régularisation sur les poids. Les solutions respectant cette contrainte se situent à l'intersection du domaine compris dans la boule et les ellipsoïdes. Cette exigence supplémentaire impose alors une réduction supplémentaire des valeurs des poids.

Soulignons que l'utilisation de la norme L_1 tend à une plus grande minimisation des poids que la norme L_2 puisque cette dernière fait intervenir le carré des paramètres. Enfin, derrière la forme circulaire de la boule L_2 , on peut déduire qu'il y a une uniformisation de la réduction alors que la forme triangulaire de la boule L_1 laisse supposer que le réseau va chercher le sous ensemble de paramètres le plus optimal pour atteindre son objectif (Sugiyama and Kawanabe, 2012). Ces différentes remarques sur la régularisation font directement écho au compromis de généralisation versus mémorisation évoqué section 3.1.1. En effet, la régularisation puisqu'elle contraint le nombre global de paramètres actifs, réduit la complexité du réseau et empêche le sur-apprentissage. C'est pourquoi imposer une régularisation fait partie du cahier des charges de l'élaboration d'un modèle d'intelligence artificielle.

Après avoir parlé des fonctions de coût et de l'incorporation des contraintes dans le cadre de l'optimisation dite "non-contrainte", ainsi que les principaux couches et outils nécessaires à la compréhension et à la construction d'un réseau de neurones, il nous reste à traiter d'un sujet fondamental qui est la minimisation en soi : l'apprentissage. Cette étape est cruciale et fondamentale puisque c'est à ce moment que les poids sont ajustés de manière à atteindre la solution optimale θ^* , solution de l'équation Eq.(3.1.2). Dans la section suivante, nous parlerons des méthodes dites de descente stochastique en introduisant brièvement quelques concepts per-

8. De manière générale on pourra considérer les pénalisation L_p tant que la norme est définie et que des précautions sont prises (notamment si $p < 1$). Le problème à optimiser s'écrira

$$\theta^* = \operatorname{argmin}_{\theta} \left[\mathcal{J} + \lambda_{L_p} \|\theta\|_{L_p} \right] \quad (3.3.7)$$

mettant de comprendre les algorithmes utilisés dans nos travaux : l'Adam et le Nadam. Nous évoquerons également les limites du premier et en quoi le deuxième a permis des résultats remarquables en terme de convergence. Les notions de moments sur lesquelles repose la principale différence entre les deux algorithmes sont décrites succinctement ; le lecteur trouvera plus de détails dans l'annexe A.

3.3.3 Algorithmes d'optimisation : du SGD au Nadam

Apprentissage : équivalence entre maximisation et minimisation - L'apprentissage d'un réseau de neurones consiste en l'optimisation de ses paramètres conformément à la tâche pour laquelle il a été construit. Cette dernière est renseignée au réseau au travers de la fonction de coût (contraintes comprises) ; elle est notée \mathcal{J} . Dans notre approche, la solution visée est une estimation ponctuelle θ^* qui maximise la distribution a posteriori (voir Eq.(3.1.3)). Du point de vue de la fonction de coût, la solution maximisant cette distribution, est exactement celle qui minimise \mathcal{J} . Par abus de langage, nous désignerons par minimisation le processus itératif permettant d'obtenir les paramètres optimaux θ^* tels que pour une tolérance δ donnée :

$$\mathcal{J}(Y_{\text{tr}}, \mathcal{F}(X_{\text{tr}}; \theta^*)) \leq \delta \quad (3.3.8)$$

avec $\mathcal{F}(X; \theta^*)$ la sortie du réseau, X_{tr} et Y_{tr} les paires d'entrées et de sorties constituant l'ensemble d'apprentissage.

Algorithme de descente et premiers raffinements - Au cours des époques d'apprentissage, l'algorithme de minimisation parcourt l'espace des paramètres Θ dans la direction réduisant la fonction de coût évaluée sur les valeurs des paramètres précédents. Il est donc nécessaire de définir le vecteur pointant vers la direction de minimisation de \mathcal{J} à chaque époque ; c'est la direction de descente. Dans les algorithmes d'optimisation, il est également conseillé de calculer le pas - suivant la direction préalablement calculée - dans l'espace des paramètres avec lequel ces derniers seront modifiés. Dans l'algorithme de descente de gradient (stochastique) initialement utilisé pour l'optimisation d'un réseau de neurone, ce pas est fixé et est désigné par le taux d'apprentissage ou *learning rate* noté η . Cette méthode basique est abrégé SGD pour Stochastique Gradient Descent elle repose sur l'algorithme Alg.(1). La stochasticité est induite par le fait que les gradients et les poids sont mis à jour à partir de patch de données *batch* plutôt que sur l'ensemble des paires $\{X_{\text{tr}}, Y_{\text{tr}}\}$. Dans l'algorithme cité ainsi que dans tous les autres, nous ne faisons pas apparaître le tirage aléatoire des patch de données pour

Algorithm 1: Descente de gradient dans sa version basique (SGD)

```

initialiser poids  $\theta_0$  ;
initialiser le taux d'apprentissage  $\eta$  ;

while  $t \leq T$  do
    |  $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$  ;
    |  $\theta_t \leftarrow \theta_{t-1} - \eta g_t$ 
end

```

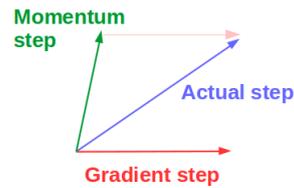


Figure 3.7 – Composition des gradient et moment pour l’établissement de la direction à suivre pour la minimisation de \mathcal{J} dans la méthode du moment classique.

éviter de surcharger les instructions. Pour conserver les notations communes à la majorité des travaux présentant les algorithmes de cette façon, nous désignons la fonction de coût par f et le gradient de la fonction de coût par rapport aux paramètres de l’itération courante (avant ajustement) par g_t . Pour les problèmes actuels, cet algorithme est souvent très lent à converger voire inefficace. La raison majeure de cette incapacité est le fait que le pas d’apprentissage ne soit pas systématiquement adapté à la situation instantanée du problème de minimisation. Cette décision ne prend en compte ni le gradient, ni la courbure (Hessienne) etc. Des tentatives d’améliorations de cette méthode consistent en la réduction (continue ou en escalier) du taux d’apprentissage fournissant une substitution aux méthodes de recherche de pas (appelées *line search*). Cette méthode - bien qu’indispensable pour permettre la convergence de la SGD - ne semble pas en adéquation avec l’idée de la recherche de pas dans les stratégies classiques d’optimisation qui consiste à *trouver* le pas idéal à franchir pour réduire sensiblement la fonction de coût.

Méthodes du deuxième ordre et moments - Des recherches d’application des méthodes du deuxième ordre comme les méthodes de Newton ou ses approximations comme la méthode BFGS (voir [Fletcher \(1987\)](#) et [Nocedal and Wright \(2006\)](#) sur ces méthodes) pour l’optimisation des NN/CNN se sont vite avérées inefficaces notamment à cause du nombre de paramètres à ajuster et de la complexité globale de la tâche. Les chercheurs se sont plutôt orientés vers une façon de prendre en compte des informations sur la courbure de la fonction de coût dans l’espace des paramètres (information du deuxième ordre) tout en restant au premier ordre. Une idée qui a radicalement changé la donne a été d’accumuler une moyenne exponentiellement décroissante des gradients calculés lors des itérations précédentes ; c’est ce que l’on appelle le moment ([Polyak, 1964](#)). Le terme fait justement écho aux “effets inertiels” dus à ces gradients précédents qui poussent (ou freinent) l’apprentissage dans la direction de g_t . De ce fait, cette accumulation simule une partie de l’information de la Hessienne ([Sutskever, 2013](#)). Quelques détails mathématiques sont évoqués dans l’annexe [A](#). Globalement la méthode du Moment classique (CM) limite les problèmes d’oscillations des méthodes SGD classiques et permet une optimisation plus efficace et plus avancée. La différence induite par le moment dans la direction d’exploration dans Θ est illustrée figure [Fig.\(3.7\)](#). La vitesse de l’“oubli”, en terme d’époque d’apprentissage, est caractérisée par un hyper-paramètre μ généralement initialisé à 0.9 (voir [A](#)). Cette constante est appelé la constante de déclin. Dans l’algorithme [Alg.\(2\)](#), le moment m_t intervient entre le calcul du gradient g_t et la mise à jour des poids. Après la composition de ces vecteurs, le réseau adapte ces paramètres selon m_t .

Cette méthode trouve son point faible dans ce qui fait justement sa force. En effet, il arrive que les vecteurs g_t et m_t pointent vers des directions opposées nuisant à la minimisation à court voire long terme, à cause des effets de mémoire. De plus, il n'existe pas de valeurs atténuant ce moment et ici encore une erreur impactera les itérations suivantes.

Pour palier à ces problèmes, la méthode de gradient accéléré de Nesterov (NAG pour Nesterov accelerated gradient) (Nesterov, 1983) a été mise au point (voir Alg.(3)). Cette méthode permet de reconnaître les directions du moment m_{t-1} potentiellement nuisible à la minimisation, avant même la mise à jour effective des paramètres. Cette prescience est rendue possible par le fait que le gradient considéré est calculé au point $\theta_t^{\text{Mom}} = \theta_{t-1} - \mu m_{t-1} \in \Theta$ avant la phase d'ajustement des points de l'époque, et non au point θ_{t-1} comme les algorithmes précédents. Plus encore, le gradient calculé sur ce point pointera toujours vers la direction de minimisation, atténuant les oscillations voire les ralentissements du CM. Notons que cette méthode n'est pas systématiquement meilleure que le CM. Elle est également moins populaire car la formulation de Nesterov impliquait une initialisation plus délicate, chronophage, par rapport à celle du CM (Sutskever, 2013), notamment dans les cas d'apprentissages stochastiques. Aujourd'hui cet algorithme est encore peu utilisé (à notre connaissance) malgré sa récente reformulation (Arjevani et al. (2015); Ruder (2016)). Nous reviendrons sur cet algorithme légèrement plus tard.

Nécessité d'un taux d'apprentissage adaptatif - Jusqu'ici, les recherches étaient principalement axées sur la direction optimale de minimisation (pour chacun des paramètres) et

Algorithm 2: Descente de gradient avec moment

```

initialiser poids  $\theta_0$  ;
initialiser taux d'apprentissage  $\eta$  ;
initialiser constante de déclin  $\mu$  ;
initialiser  $m_t = 0$  ;

while  $t \leq T$  do
     $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$  ;
     $m_t \leftarrow \mu m_{t-1} + \eta g_t$  ;
     $\theta_t \leftarrow \theta_{t-1} - m_t$  ;
end

```

Algorithm 3: Descente de gradient avec moment accéléré de Nesterov (Nesterov, 1983)

```

initialiser poids  $\theta_0$  et  $m_t = 0$  ;
initialiser taux d'apprentissage  $\eta$  et la constante de déclin  $\mu$  ;
while  $t \leq T$  do
     $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1} - \mu m_{t-1})$  ;
     $m_t \leftarrow \mu m_{t-1} + g_t$  ;
     $\theta_t \leftarrow \theta_{t-1} - \eta m_t$  ;
end

```

peu sur le pas optimal à effectuer dans l'espace Θ pour chaque paramètre (Ruder, 2016). Si les moments permettent de meilleurs apprentissages, les architectures toujours plus complexes et profondes ont fait resurgir le problème de l'adaptation des poids en particulier pour les CNN. Pour comprendre la limitation imposée par le fait de ne pas adapter les taux d'apprentissage selon les directions est la suivante : lorsqu'un CNN apprend un pattern dans les données, le gradient dans l'espace des paramètres dans la direction ayant favorisé son apprentissage est fort. De manière générale, pour apprendre des caractéristiques plus complexes au sein du réseau, il faudra au préalable réduire les gradients dans les directions déjà apprises et amplifier les autres. De cette manière, l'information apprise demeure inchangée et le moment (dans le cas des CM et NAG) encouragera les directions dans lesquels d'autres patterns pourront être appris. Il apparaît donc nécessaire de moduler les taux d'apprentissage en fonction des dynamiques d'apprentissage afin de permettre un apprentissage plus stable et plus profond. Comme nous l'avons déjà abordé, au cours de son apprentissage le réseau déchiffre des patterns de plus en plus complexes à mesure que le réseau s'approfondit. Mais pour ne pas que ces caractéristiques soient apprises au dépend des autres, il est nécessaire de moduler les vitesses d'apprentissages selon les différentes directions (propre à chaque paramètre).

Méthodes d'adaptation du taux d'apprentissage - Geoff Hinton présente le RMSProp (Root Mean Square Prop) (Tieleman and Hinton, 2012). C'est une méthode basée sur un algorithme mis au point peu de temps avant leurs travaux à savoir l'Adagrad (Duchi et al., 2011). Dans ce dernier algorithme, le taux d'apprentissage est divisé par la norme L_2 de tous les gradients précédents, additionnés en amont de la mise à jour des poids. L'opération supplémentaire consiste à calculer au temps t

$$n_t \leftarrow n_{t-1} + g_t^2 \quad (3.3.9)$$

avec g_t^2 le vecteur des carrés des gradients associées à chaque poids. De façon plus rigoureuse, on peut construire une matrice diagonale $G_{t;i,j}$ définie comme

$$G_{t;i,j} = \begin{cases} \sum_{\tau=1}^t g_{\tau;i}^2 & i = j, \\ 0 & \text{Sinon.} \end{cases} \quad (3.3.10)$$

Algorithm 4: RMSProp (Tieleman and Hinton, 2012)

initialiser poids θ_0 avec une stratégie d'initialisation;
initialiser taux d'apprentissage η , la constante de déclin ν ;
initialiser $n_t, m_t = 0$;

while $t \leq T$ **do**

$g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$;
 $n_t \leftarrow \nu n_{t-1} + (1 - \nu) g_t^2$;
 $\theta_t \leftarrow \theta_{t-1} - \eta g_t / (\sqrt{n_t} + \varepsilon)$;

end

Pour éviter toute confusion, nous écrivons la mise à jour du poids $\theta_{t;i}$

$$\theta_{t;i} \leftarrow \theta_{t-1;i} - \eta \frac{g_{t;i}}{\sqrt{G_{t;ii} + \varepsilon}} \quad (3.3.11)$$

Cette méthode permet d'adapter le taux d'apprentissage par rapport aux gradients associés à chaque paramètre : si l'apprentissage est rapide sur un paramètre (accumulation de forts gradients), le taux d'apprentissage est progressivement réduit ; si l'apprentissage est lent selon un autre, il est accéléré (idem). Le fait de normaliser le gradient à chaque époque selon s'il a significativement changé au cours de l'apprentissage permet d'apprendre plus rapidement les patterns les plus rares comme les plus communs (Dozat, 2016). Adagrad permet des progrès en terme de temps de convergence par rapport aux méthodes au temps d'apprentissage non adaptatif ; cependant, le fait d'accumuler la norme des gradients précédents (forcément positive) sans modulation empêche petit à petit l'apprentissage du réseau tant le rapport taux d'apprentissage / gradient cumulé sera proche de zéro pour chacun des poids.

C'est principalement ici que le RMSProp (ainsi que l'algorithme Adadelta (Zeiler, 2012)) se distingue du Adagrad. La norme L_2 est remplacée par une moyenne (exponentiellement) déclinante de la norme des gradients précédents, paramétrisée par ν le facteur d'oubli des normes des gradients précédents. La grandeur n_t est alors mise à jour de la façon suivante

$$n_t \leftarrow \nu n_{t-1} + (1 - \nu) g_t^2 \quad (3.3.12)$$

Puisque l'impact des gradients précédents décroît de façon exponentielle, le rapport taux d'apprentissage / gradient cumulé ne tend plus irrémédiablement vers 0 et permet un apprentissage théoriquement constant. L'algorithme du RMSProp est présenté Alg.(4).

Algorithmes combinant adaptabilité et moment - Jusqu'ici on distingue deux types d'améliorations du SGD : l'utilisation de moment pour ajuster la direction d'apprentissage pour chaque poids, et l'utilisation des gradients pour ajuster la magnitude du changement des paramètres selon la direction de descente. Kingma and Ba (2014) et Dozat (2016) ont eu l'idée de combiner ces deux méthodes. Le premier combine utilise le moment classique pour optimiser la direction de descente selon les gradients précédents avec l'idée d'adaptabilité du RMSProp. C'est l'algorithme Adam qui est de loin le plus utilisé dans la littérature ML et ML/MDF. Dans les instructions de son algorithme Alg.(5), on retrouve l'incrément de moment classique ainsi que la moyenne pondérée des gradients carré de chaque poids propre au RMSProp. Les grandeurs \hat{m}_t et \hat{n}_t sont des étapes rajoutées pour faciliter la lecture de l'algorithme.

Le Nadam combine l'approche accélérée de Nesterov avec le RMSProp. Son algorithme Alg.(6) est basé sur une réécriture du moment de Nesterov et du Adam détaillée dans Dozat (2016). Il profite des avantages du moment de Nesterov et accélère souvent la convergence par rapport au Adam, tout en étant plus performant (Ruder (2016), Dozat (2016) et Tato and Nkambou (2018)), il n'est pas très présent dans la bibliographie à notre connaissance.

Durant nos travaux, nous avons rencontré plusieurs difficultés dans l'entraînement des CNN. Après avoir essayé de comprendre l'origine de cette défaillance en complexifiant le modèle, en testant la saturation du réseau en changeant les fonctions d'activation, en testant différentes

stratégies d'initialisation ou encore en réduisant le taux d'apprentissage initial puis planifier la réduction du taux d'apprentissage au fur et à mesure de l'avancement de l'apprentissage, il s'est avéré que la minimisation du réseau n'était pas suffisante pour fournir des résultats satisfaisants sur l'ensemble des champs reconstitués aux niveaux tensoriel et à plus forte raison vectoriel et scalaire. Des méthodes d'exploration des hyper-paramètres comme le *grid search* et le *random search* (Goodfellow et al., 2016) confirmèrent que l'optimiseur Adam pourtant le plus populaire n'était pas adapté pour ce problème. Puis il s'est avéré qu'en utilisant l'optimiseur Nadam les résultats furent véritablement meilleurs et que la convergence du réseau se faisant en 10 à 20 fois moins d'itérations qu'avec l'optimiseur le plus répandu (presque à l'unanimité) dans la bibliographie (voir section 4.1). Nous avons ensuite essayé cet optimiseur pour nos travaux de reconstructions et encore une fois les résultats étaient supérieurs. Cependant notons que le réseau est très sensible à l'initialisation du taux d'apprentissage et c'est ce qui peut expliquer le manque de popularité de la méthode.

Il serait prétentieux de prétendre comprendre pourquoi l'algorithme Nadam permet une différence de résultat aussi grande qu'en utilisant l'algorithme Adam. Intuitivement, il semblerait que l'Adam se coince très vite dans un minimum local et que le moment classique ne soit pas suffisant pour permettre à l'optimisation de repartir dans la bonne direction. Une raison supplémentaire ou complémentaire pourrait provenir du fait que la Hessienne sous jacente possèdent des valeurs propres très différentes rendant inefficace le CM (voir Annexe A et (Sutskever, 2013, chap.8)). Le moment de Nesterov est plus robuste face à ces deux problèmes notamment par le fait que la minimisation se fait dans une étape virtuelle permettant de se sortir des minima locaux.

Conclusion du chapitre - Au cours de ce chapitre, nous avons introduit les concepts de réseaux de neurones ainsi que de la particularité des réseaux de neurones à convolutions. Nous avons également exprimé le but d'un réseau de neurones dans le contexte d'une prédiction ponctuelle. Par la suite, nous avons détaillé les différents éléments qui constituent le «comment» atteindre un tel but. La première étape, peut être la plus primordiale, eut été de présenter les propriétés des différentes fonctions d'activation classiques, et moins classiques à l'image de la fonction Mish, pour mieux s'orienter dans le vaste espace des hyper-paramètres. Nous avons également discuté des fonctions de coût les plus classiques et les plus utilisées, ainsi que de

Algorithm 5: Adaptive moment estimation (Adam) (Kingma and Ba, 2014)

```

initialiser poids  $\theta_0$  ;
initialiser taux d'apprentissage  $\eta$  ;
initialiser constante de déclin  $\mu$  ;
initialiser  $n_t, m_t = 0$  ;

```

```

while  $t \leq T$  do

```

```

     $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$  ;
     $m_t \leftarrow \mu m_{t-1} + (1 - \mu) g_t$  ;
     $\hat{m}_t \leftarrow m_t / (1 - \mu)$  ;
     $n_t \leftarrow \nu n_{t-1} + (1 - \nu) g_t^2$  ;
     $\hat{n}_t \leftarrow n_t / (1 - \nu)$  ;
     $\theta_t \leftarrow \theta_{t-1} - \eta \hat{m}_t / (\sqrt{\hat{n}_t} + \varepsilon)$  ;

```

```

end

```

Algorithm 6: Nesterov adaptative moment estimation (Nadam) (Dozat, 2016)

initialiser poids θ_0 ;
 initialiser taux d'apprentissage η ;
 initialiser constante de déclin μ ;
 initialiser $n_t, m_t = 0$;

while $t \leq T$ **do**

$g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$;
 $\hat{g}_t \leftarrow g_t / (1 - \prod_{i=1}^t \mu_i)$;
 $m_t \leftarrow \mu m_{t-1} + (1 - \mu)g_t$;
 $\hat{m}_t \leftarrow m_t / (1 - \prod_{i=1}^{t+1} \mu_i)$;
 $n_t \leftarrow \nu n_{t-1} + (1 - \nu)g_t^2$;
 $\hat{n}_t \leftarrow n_t / (1 - \nu)$;
 $\bar{m}_t \leftarrow (1 - \mu)\hat{g}_t + \mu_{t+1}\hat{m}_t$;
 $\theta_t \leftarrow \theta_{t-1} - \eta \bar{m}_t / (\sqrt{\hat{n}_t} + \varepsilon)$;

end

la possibilité d'ajouter des contraintes au problème de minimisation globale non contrainte, que nous cherchons à résoudre dans la phase d'apprentissage du réseau. Ce paradoxe trouve sa solution dans une généralisation des multiplicateurs de Lagrange et permet de construire des fonctions d'objectifs fournissant beaucoup plus d'informations sur l'état des paramètres du réseau par rapport au but final. C'est également un moyen de renforcer l'apprentissage en lui garantissant un caractère plus universel, plus général. Enfin, nous avons discuté de l'étape au cœur de ce dit apprentissage, à savoir la minimisation de la fonction de coût par des méthodes de descentes de gradients du premier ordre.

Le chapitre suivant aborde la question de ce qui existe et qui fonctionne dans la littérature, puis présente dans les moindres détails le modèle que nous avons sélectionné pour mener à bien nos travaux. À la suite de cela, nous évaluerons la qualité de l'apprentissage et de la capacité de généralisation du CNN entraîné.

Chapitre 4

Élaboration d'une intelligence artificielle pour la mécanique des fluides

Les deux chapitres précédents ont posé les bases ainsi que les définitions fondamentales qui sont au cœur de la thèse. Ce chapitre est entièrement orienté sur la combinaison intelligence artificielle et mécanique des fluides. Nous commençons par présenter les grands axes de la recherche couplant méthodes d'IA et mécanique des fluides (MDF). La variété et la quantité des articles a imposé un choix, et notre revue bibliographique sera à compléter avec les articles spécialisés (Brunton et al., 2020; Duraisamy, 2020; Duraisamy et al., 2019; Schmidhuber, 2015; Xiao and Cinnella, 2019). Dans un second temps, nous présenterons brique par brique la méthode et le cadre que nous avons conçus pour mener à bien notre défi. Enfin après avoir détaillé une partie de l'optimisation des hyper-paramètres, nous évaluerons l'apprentissage ainsi que les prédictions sur l'ensemble de test conformément aux vérifications d'usage en intelligence artificielle.

4.1 Historique de l'évolution du couplage Mécanique des Fluides et réseau de neurones

4.1.1 Injection de l'IA dans l'équation de fermeture des modèles RANS

Pour l'essentiel, les articles présentés ici utilisent la méthode d'inférence Bayésienne dont le cœur est rappelé dans la section 3.1.1 et notamment à travers le Théorème de Bayes Eq.(3.1.3). Dans le paragraphe suivant nous illustrons cette méthode et sa généralisation par machine learning ; nous présentons également les idées principales d'articles utilisant le couplage inférence Bayésienne et ML. Un des procédés les plus utilisés pendant la première moitié de la dernière décennie était le couplage Inférence Bayésienne (IBay) / IA. De manière générale, on peut représenter par $\mathcal{L}(x, t)$ l'opérateur non linéaire regroupant les équations de Navier-Stokes dans leur formulation exacte. Pour minimiser les temps de calculs, les chercheurs ont construit des

modèles de simulation numérique dont les équations - inexactes - sont représentées par $\tilde{\mathcal{L}}(x, t; \beta)$. Le tenseur β symbolise toute la physique que le modèle ne considère que partiellement voire pas du tout. La modélisation de cette physique manquante peut se faire par des formules abstraites ou algébriques comme c'est le cas des modèles RANS ou LES non "augmentés" par assimilation de données. L'inférence Bayésienne, elle, cherche à calculer la distribution *a posteriori* de la grandeur β telle que les résultats du modèle soient les plus proches possibles des données issues des équations exactes ou des expériences ; pour des raisons de faisabilité, il est d'usage d'utiliser l'estimation ponctuelle de type MAP (voir 3.1.1 et (3.1.5)). Le tenseur β obtenu est alors défini sur l'espace. À condition de définir les équations et de trouver un moyen efficace de calculer les gradients nécessaires et parfois les dérivées successives dans le processus d'optimisation (voir Nocedal and Wright (2006)), il est possible d'inférer le tenseur β sur une grande quantité de problèmes (voir Tarantola (2005), Aster et al. (2011) entre autres). Bien souvent, calculer les dérivées premières et secondes d'un problème d'inférence impliquant la mécanique des fluides est un procédé coûteux (en temps de calcul) et le tenseur β issu de l'IBay au temps t sera propre à l'écoulement, à cette même itération, et donc non-applicable à d'autres itérations ou simulations. Ce coût et le fait de devoir définir une distribution a priori (*prior*) cohérente avec le problème d'inférence étudié (voir les équations (3.1.4) et (3.1.2), ainsi que les discussions autour de la nécessité de définir une prior convenable ; voir également Tarantola (2005), Bishop (2006) et Parish and Duraisamy (2016) entre autres) a quelque peu stoppé l'élaboration de modèles de MDF utilisant l'IBay jusque récemment. En effet avec l'essor des performances des ordinateurs et le regain d'intérêt pour l'apprentissage automatique, plusieurs articles font état de combinaisons fructueuses entre IBay et IA, dans le cadre de la MDF. L'IA permet de généraliser les tenseurs β et d'en constituer une fonctionnelle β^{ML} en fonction de grandeurs propres et/ou caractéristiques des écoulements sur lesquels des IBay successives ont été appliquées ; cette formulation est écrite Eq.(4.1.1) dans laquelle les η_i , $i \in \{1, n\}$ représentent les grandeurs sélectionnées pour l'apprentissage de \mathcal{F} la fonctionnelle recherchée.

$$\beta^{\text{ML}} = \mathcal{F}(\eta_1, \eta_2, \dots, \eta_n) \quad (4.1.1)$$

Ce couplage est au cœur de plusieurs articles (Duraisamy, 2016; Duraisamy and Durbin, 2014; Duraisamy et al., 2015; Singh et al., 2017b). Les auteurs cherchent à corriger les modèles RANS notamment Wilcox (Wilcox et al., 1998), Spalart-Almaras (Spalart and Allmaras, 1992) englobant les modèles à une et deux équations existants. Suivant les équations de ces modèles, les auteurs cherchent à inférer une ou plusieurs grandeurs apparaissant dans ces équations (facteur d'intermittence γ , coefficient de portée C_l etc.). Pour se faire, les auteurs "augmentent" ou "informent" les modèles avec des données DNS ou expérimentales. L'inférence se fait alors entre ces dernières et les solutions du modèle. Comme dans de nombreux articles, les dérivées sont calculées en utilisant les adjoints (Lakshminarayan and Duraisamy (2015); Singh and Duraisamy (2016)); il est également possible de calculer les gradients en utilisant la méthode exposée dans Mons et al. (2016). Pour la généralisation des résultats de l'IBay, les auteurs utilisent principalement les réseaux de neurones (NN) et plus rarement les Processus Gaussiens (GP en anglais)¹.

1. Premièrement, les résultats sont plus précis avec les NN. Deuxièmement, la formulation même des GP interroge sur la notion d'apprentissage et de prédiction, voir notamment (Rasmussen and Williams, 2006, chap.2).

4.1.2 Prédiction de la partie anisotropique du tenseur de Stress (RANS) et injection d'invariance

Quelques mots sur les équations RANS - L'équation de champ moyen pour les modèles RANS s'écrit

$$\frac{\partial \langle U_i \rangle}{\partial t} + \langle U_j \rangle \frac{\partial \langle U_i \rangle}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \langle P \rangle}{\partial x_i} - \frac{\partial}{\partial x_j} \left(\langle u'_i u'_j \rangle + \nu \frac{\partial \langle U_i \rangle}{\partial x_j} \right) \quad (4.1.2)$$

$$\frac{\partial \langle U_i \rangle}{\partial x_i} = 0 \quad (4.1.3)$$

avec la décomposition de Reynolds des champs turbulent s'écrivant

$$\xi = \langle \xi \rangle + \xi' \quad (4.1.4)$$

Pour les trois équations, la notation $\langle \xi \rangle$ représente le champ de moyen et ξ' le champ fluctuant. L'équation Eq.(4.1.2) s'obtient en injectant cette décomposition dans les équations Eqs.(2.1.10a) et (2.1.10b), puis en moyennant ces équations. La difficulté dans les méthodes RANS est la modélisation du tenseur $\tau_{ij}^{\text{RANS}} = \langle u'_i u'_j \rangle$. Ce terme décrit l'action du champ fluctuant (non résolu par le modèle) sur le champ moyen. Comme pour le cas LES, essayer de calculer théoriquement τ_{ij}^{RANS} conduit à une cascade d'équations contenant chacune des inconnues nécessitant une équation supplémentaire. La modélisation entend donc fournir une équation de fermeture au système Eqs.(4.1.2) et (4.1.3) évitant la cascade.

Modèle de viscosité turbulente généralisé (GEVM) - La majorité des articles se basent sur l'hypothèse de Boussinesq proposant une viscosité turbulente ν_t en analogie à la viscosité moléculaire. À l'origine de cette hypothèse, on décompose tenseur τ_{ij}^{RANS} en une partie isotropique et une anisotropique (Pope, 2001), elles sont repérées par deux couleurs différentes dans l'équation suivante :

$$\tau_{ij}^{\text{RANS}} = \underbrace{\frac{2}{3} k \delta_{ij}}_{\text{Partie isotropique}} + \underbrace{b_{ij}}_{\text{Partie anisotropique}} \quad (4.1.5)$$

Dans la forme initiale de l'hypothèse de Boussinesq, la partie anisotropique (terme en bleu dans l'équation précédente), notée b_{ij} s'écrit :

$$b_{ij} = -2\nu_t \langle S_{ij} \rangle \quad (4.1.6)$$

Par la suite nous ferons référence à ce modèle par l'abréviation LEVM (pour Linear Eddy Viscosity Model). Souvent, la partie anisotropique nécessite une formulation plus complexe, notamment celle proposée par Pope (1975) utilisant les bases intègres. Dans la suite de cette section, les tenseurs (turbulents) sont tous moyennés, sauf précision du contraire ; nous nous affranchissons alors de la notation moyennée $\langle \bullet \rangle$. On peut écrire :

$$b_{ij} = \sum_{n_1}^{10} g^{(n)}(\Omega_1, \dots, \Omega_5) T_{ij}^{(n)} \quad (4.1.7)$$

où $g^{(n)}$ sont des coefficients définis en fonction des invariants Ω_i , $i \in [1, 5]$ qui sont (Ling et al., 2016a; Pope, 1975)

$$\Omega = \{ \text{Tr}(\mathbf{S}^2), \text{Tr}(\mathbf{R}^2), \text{Tr}(\mathbf{S}^3), \text{Tr}(\mathbf{R}^2\mathbf{S}), \text{Tr}(\mathbf{R}^2\mathbf{S}^2) \}$$

Les tenseurs considérés \mathbf{S} et \mathbf{R} sont respectivement les tenseurs de déformations et de rotations solides adimensionnés par l'énergie cinétique et la dissipation :

$$S_{ij} = \frac{1}{2} \left(\frac{\partial \langle U_i \rangle}{\partial x_j} + \frac{\partial \langle U_j \rangle}{\partial x_i} \right) \quad R_{ij} = \frac{1}{2} \left(\frac{\partial \langle U_i \rangle}{\partial x_j} - \frac{\partial \langle U_j \rangle}{\partial x_i} \right) \quad (4.1.8)$$

T est composée de 10 tenseurs dans le cas tridimensionnel. Les 10 tenseurs proposés par Pope (1975) et repris par Ling et al. (2016b) sont :

$$\begin{aligned} \mathbf{T}^{(1)} &= \mathbf{S} & \mathbf{T}^{(6)} &= \mathbf{R}^2\mathbf{S} + \mathbf{S}^2\mathbf{R} - \frac{2}{3} \mathbf{I} \cdot \text{Tr}(\mathbf{S}\mathbf{R}^2) \\ \mathbf{T}^{(2)} &= \mathbf{S}\mathbf{R} - \mathbf{R}\mathbf{S} & \mathbf{T}^{(7)} &= \mathbf{R}\mathbf{S}\mathbf{R}^2 - \mathbf{R}^2\mathbf{S}\mathbf{R} \\ \mathbf{T}^{(3)} &= \mathbf{S}^2 - \frac{1}{3} \mathbf{I} \cdot \text{Tr}(\mathbf{S}^2) & \mathbf{T}^{(8)} &= \mathbf{S}\mathbf{R}\mathbf{S}^2 - \mathbf{S}^2\mathbf{R}\mathbf{S} \\ \mathbf{T}^{(4)} &= \mathbf{R}^2 - \frac{1}{3} \mathbf{I} \cdot \text{Tr}(\mathbf{R}^2) & \mathbf{T}^{(9)} &= \mathbf{R}^2\mathbf{S}^2 + \mathbf{S}^2\mathbf{R}^2 - \frac{2}{3} \mathbf{I} \cdot \text{Tr}(\mathbf{S}^2\mathbf{R}^2) \\ \mathbf{T}^{(5)} &= \mathbf{R}\mathbf{S}^2 - \mathbf{S}^2\mathbf{R} & \mathbf{T}^{(10)} &= \mathbf{R}\mathbf{S}^2\mathbf{R}^2 - \mathbf{R}^2\mathbf{S}^2\mathbf{R} \end{aligned}$$

L'écriture (4.1.7) constitue le modèle de viscosité turbulente généralisée (GEVM en anglais).

Tensor basis neural network (TBNN) - Ce modèle (GEVM) a été utilisé pour la construction d'un réseau de neurones particulier par Ling et al. (2016b) et Kutz (2017) dont l'intérêt réside dans son architecture. Le réseau de neurones possède deux couches d'entrées : une première par laquelle les invariants sont fournis au réseau, puis plus loin dans le réseau, une deuxième couche d'entrée permet d'injecter les valeurs des tenseurs de la base intègre. Pour obtenir la sortie globale du réseau, les auteurs ont imaginé une connexion de type multiplication élément par élément entre la sortie intermédiaire du réseau au niveau de la deuxième entrée, et les entrées de cette dernière. Cela force à obtenir une sortie de la même forme que l'équation Eq.(4.1.7). Comme le précise la théorie, tout tenseur pouvant s'écrire comme une combinaison linéaire des éléments de la base intègre satisfait automatiquement les invariances galiléennes et en particulier celles des rotations du groupe $\mathcal{SO}(3)$. L'utilité du réseau est donc double ; dans un premier temps il permet de calculer les coefficients $g^{(n)}$ assurant le mapping entre la modélisation GEVM et les données exactes (DNS et/ou expérimentales). Dans un deuxième temps, il pondère ces coefficients à partir de paramètres et fournit un tenseur anisotrope b_{ij} qui vérifie les lois les plus fondamentales de la Mécanique des Fluides. Que ce soit l'architecture ou la démarche basée sur les bases intègres, ce travail a inspiré nombre d'articles, chacun proposant une amélioration au niveau de la qualité des résultats Wang et al. (2017), Akolekar et al. (2019) et Wu et al. (2018). Dans ces articles, il est souligné que le choix de la base intègre n'est peut être pas la plus optimale, comme pressenti dans l'article pionnier. À la place, ils proposent d'appliquer la théorie de Pope (1975) pour construire une base intègre comprenant 47 éléments (ils rajoutent également trois éléments pour informer l'IA des caractéristiques

géométriques du domaine, et autres). Se faisant, ils prouvent qu'il est possible de prendre en compte les écoulements secondaires comme les gradients de pression inverses, et permet de s'affranchir de l'hypothèse d'équilibre entre production et dissipation d'énergie cinétique, au cœur des modèles RANS comme $k-\varepsilon$. Pour assurer toujours plus de capacité de généralisation et des invariances par rotations de l'IA, ils ne prédisent pas directement le tenseur b_{ij} mais établissent un mapping entre les positions des vecteurs propres du tenseur obtenu par le RANS exprimées dans un repère à quatre dimension (unité de quaternion) et celles nécessaires pour améliorer cette prédiction. Ils prédisent en quelque sorte l'écart existant entre certaines propriétés de la modélisation et celles des données de références pour pouvoir s'approcher le plus possible d'un tenseur exact, assurant les bonnes propriétés, à commencer par les invariances propres à la mécanique. Toujours dans la veine du TBNN, Fang et al. (2018) propose une interprétation physique des résultats du TBNN. Ils montrent notamment que ce réseau est affecté par la limitation du modèle (en tant que tel) GEVM en mettant en évidence un comportement très intéressant du réseau au cours de son apprentissage. En effet, le réseau entraîné à partir des données exactes essaie de corriger les erreurs constitutives du GEVM en activant certaines connexions qui théoriquement aurait du être éteintes (à cause de la géométrie du domaine). Cette correction artificielle peut amener à une pollution des résultats. Une façon de contourner ce problème est d'ajouter de la complexité au modèle en augmentant le nombre de couches cachées. Ceci confèrera la flexibilité nécessaire pour permettre au réseau de dénouer certaines relations, isoler les comportements indésirables et obtenir une meilleure performance globale.

Injection d'invariance - Le fait d'induire les invariances dans l'IA peut passer par son architecture, mais également par les données fournies lors de l'apprentissage. C'est l'idée de Dieleman et al. (2015); Kaandorp and Dwight (2018); Ling et al. (2016a) ou encore Al Sayed (2011). Des efforts dans ce sens peuvent être faits dans le domaine de l'apprentissage par renforcement, des réseaux récurrents ou des CNN en utilisant notamment les méthodes proposées dans les articles Cohen and Welling (2016); Gens and Domingos (2014); Kandi et al. (2019); Li et al. (2018) dont la majorité se basent sur la théorie des groupes.

Dans la totalité des travaux que nous avons présentés (en dehors des quatre derniers qui ne

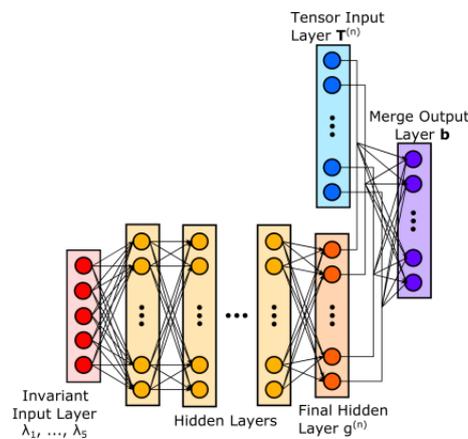


Figure 4.1 – TBNN proposé par Ling et al. (2016b)

traitent pas de MDF), l'IA est utilisée pour effectuer des prédictions point par point. Les TBNN sont construits à partir de réseaux à connexions denses s'approchant de l'appellation "profonds" avec une dizaine de couches contenant une dizaine de neurones voire moins. Les réseaux impliqués dans les tâches de généralisation de l'inférence étaient plus profonds avec des réseaux contenant peu de couches cachées (deux voire trois), mais beaucoup de neurones par couche (en centaine). Au niveau des hyper-paramètres, les taux d'apprentissage ont été optimisés; dans la majorité des cas l'algorithme SGD classique est utilisé, parfois le Adam. Les couches des réseaux TBNN sont activées par la fonction leaky-relu (voir 3.2); dans les cas d'inférence elles le sont par des sigmoïdes.

La section suivante présente des travaux dans lesquels les auteurs contournent le problème de fermeture en fournissant directement le tenseur des contraintes, ou en traitant le champ de vitesse filtré dans le cas de la LES.

4.1.3 Travaux effectués dans le contexte de la LES

Plusieurs articles se concentrent sur la prédiction du tenseur de contraintes sous-mailles symbolisant les interactions entre échelles non résolues et les échelles résolues/non résolues, dans le contexte de la LES. [Zhu et al. \(2019\)](#) modélisent ce tenseur en fonction du champ de vitesse filtré puis du gradient du champ de vitesse filtré. Ils utilisent un réseau à une seule couche cachée impliquant 120 neurones connectés de manières denses avec les couches d'entrées et de sorties. Ils explorent la possibilité d'entraîner le réseau avec plusieurs tailles de filtres. Point à point, ils montrent que l'utilisation du gradient en entrée est plus à même de fournir des résultats dépassant les performances du modèles de Clark - en terme de corrélation de Pearson. Les auteurs ont effectué leurs analyses au niveau tensoriel uniquement. Les travaux de [Pal \(2019\)](#) cherchent à généraliser le modèle de Smagorinsky dynamique (SDM) (définis par [\(Pope, 2001\)](#) entre autres) par un réseau de neurones densément connectés. Ils nomment leur technique viscosité turbulente intelligente (INU en anglais) et prouvent que pour les cas considérés, le modèle est plus efficace que le SDM qualitativement et quantitativement. D'autres travaux ([Sarghini et al., 2003](#)), calculent le coefficient de Smagorinsky C_s pour faire correspondre la modélisation de ce modèle avec ceux de la DNS. On parlera également de modèle de Smagorinsky intelligent. Les deux derniers cas sont des exemples d'application de réseaux de neurones en substitution intégrale du couplage IBay/IA. Ils illustrent également le fait que l'on peut utiliser l'IA pour augmenter un modèle déjà existant.

Il existe également des travaux utilisant la technique de déconvolution approximative (ADM). Cette méthode permet de reconstruire le champ de vitesse *total* à partir du champ filtré, puis, à l'aide de ce dernier, d'obtenir le tenseur τ conformément à l'équation Eq.(2.3.5) (voir [Ferziger et al. \(2002\)](#); [Layton and Rebholz \(2012\)](#) pour plus de détails sur cette méthode). [Maulik and San \(2017\)](#) utilisent l'extrême machine learning (ELM, voir [Huang et al. \(2004\)](#)) pour généraliser l'ADM, sans informer sur la nature du filtrage des données. Les articles [Maulik et al. \(2019\)](#); [Yuan et al. \(2020\)](#) développent le même principe de NN appliqué à l'ADM, en utilisant cette fois des réseaux plus profonds et plus conventionnels avec des activations ReLU. Jusqu'à présent tous les travaux les plus innovants les uns que les autres n'impliquaient qu'un seul de type de réseaux de neurones - les réseaux à connexions denses. Ces réseaux semblent

adaptés pour ce genre de problème et fournissent également de très bons résultats. Les articles que nous présentons à présent considèrent des données multi-dimensionnelles.

Les travaux de Beck et al. (2019) (Kurz and Beck, 2020) ont mis au point une méthode proche de celle que nous allons présenter. Ils concernent la turbulence de fluides compressibles et essaient de prédire le terme de contraintes sous-maille inconnu. Plus précisément, ils prédisent la divergence de ce tenseur tel qu'elle apparaît dans les équations d'Euler (équivalent de Navier-Stokes pour les fluides compressibles). Ils utilisent en entrée les champs de vitesse et des grandeurs connues, présents dans cette même équation. Les prédictions se font en volume en découpant le volume en cubes, sans préciser la logique de décomposition. Les premiers (Beck et al., 2019) comparent les méthodes basées sur les Perceptron multi-couches (MLP, autre nom du ANN densément connecté), les réseaux à convolutions et des réseaux avec des couches récurrentes. Les premiers décomposent les cubes en deux parties : l'intérieur du cube et les points les plus extérieurs c'est-à-dire constituant la surface du cube. Dans toutes ces configurations, les corrélations de Pearson présentées dans les différents tableaux se situent en moyenne à 35%. Suivant la même méthode mais élaborant un modèle plus profond Kurz and Beck (2020) explorent la puissance des réseaux récurrents. Ils utilisent notamment des unités spécifiques appelées LSTM - long short term memory - composées de plusieurs fonctions d'activations. Elles permettent la transition et la combinaison de l'information des itérations courantes et passées. Chaque sortie intermédiaire d'une couche cachée récurrente, est considérée comme entrée pour la couche cachée suivante. La sortie de la dernière couche cachée intermédiaire est fournie à la couche dense suivante (dans le cas de l'article). Dans cet article, l'IA entraînée exprime de très fortes capacités d'apprentissage et de généralisation. Enfin, les auteurs apprennent à prédire les composantes du vecteur de divergence du tenseur en question selon différentes tailles de filtre. Les corrélations de Pearson sont très fortes : plus de 99%.

Notre approche partage des similarités avec plusieurs articles et se démarque sur différents points. Nous construisons une IA capable de prédire les six composantes du tenseur des contraintes à partir des composantes du champ de vitesse filtré dans le cadre de la LES. Il s'est avéré que prédire l'ensemble du champ était 1- chronophage, aussi bien pour la génération de données que pour l'entraînement, et 2- revenait à limiter la portance de notre méthode aux simulations qui auront la même résolution que les simulations considérées lors de l'entraînement. Pour solutionner ces points, nous avons décidé de découper le domaine global en cubes dont la taille est étudiée en fonction du κ_C le nombre d'onde de coupure. Nous construisons également une fonction de coût qui englobe plus d'informations que les fonctions de coût considérés dans la presque totalité des articles mentionnés ; nous utilisons une fonction d'activation inédite amenant à de meilleurs résultats que les combinaisons renormalisation/ReLU, sans avoir besoin d'utiliser la *batch normalization*. Nous présentons également toutes les caractéristiques de l'architecture U-net et expliquons en quoi elle correspond à notre problème, aussi bien au niveau de ses forces intrinsèques que de son utilisation dans la littérature. Nous nous baserons notamment sur l'article Lapeyre et al. (2019) que nous développerons spécialement dans la section 4.2.2. Enfin et surtout, nous avons construit un modèle applicable à des écoulements plus de 100% plus turbulents que celui utilisé pour l'apprentissage laissant penser que l'intelligence artificielle a trouvé une relation généralisable (peut être même intrinsèque) liant \bar{U} à τ^{SGS} . Les caractéristiques du modèle et ses capacités d'apprentissage et de généralisation sont au cœur des sections suivantes. L'application aux autres écoulements est laissée pour le chapitre suivant.

4.2 Cadre de travail : ensemble d'entrée-sorties et réseau U-net

Comme nous l'avons vu, la majorité des articles scientifiques qui utilisent une combinaison des méthodes traditionnelles de la résolution des écoulements turbulents avec les algorithmes d'intelligence artificielle promeuvent l'utilisation de données 1D c'est-à-dire sous forme de vecteurs, dont les composantes représentent les grandeurs turbulentes évaluées selon un schéma de discrétisation proche de ceux que l'on retrouve dans les méthodes numériques telles que les volumes finis. Nous avons plutôt opté pour une discrétisation en volume pour pouvoir utiliser au maximum les capacités des réseaux à convolutions. Dans la section suivante, nous présentons la construction des entrées et des sorties du réseau.

4.2.1 Élaboration des bases de données

Dilemme information vs. faisabilité - L'élaboration du procédé de construction des entrées et des sorties entend résoudre un dilemme. Nous avons vu dans le chapitre 2.2.2 que les interactions entre modes turbulents ne sont pas locales dans l'espace de Fourier, ce qui laisse sous entendre que des structures tourbillonnaires de taille et d'énergie cinétique (vitesse) différentes rentrent en jeu pour la description d'un phénomène présent dans le champ de vitesse. Pour s'assurer de couvrir l'ensemble de l'événement et d'obtenir le tenseur des contraintes résultant de toutes ces interactions, il est plus intuitif de considérer l'ensemble du champ de vitesse. Si le nombre de points du domaine tri-périodique que nous considérons est donné par N , alors toutes les grandeurs en entrée et en sortie seront composées de N^3 points. La taille du domaine cubique considéré est donnée par $L = N\Delta$: le nombre de points N multiplié par le pas de discrétisation choisi Δ . Comme nous l'avons abordé section 2.2.2, pour éviter les problèmes de sous échantillonnage et de manquer d'information sur les effets visqueux aux échelles de Kolmogorov, les méthodes DNS imposent des discrétisations très fines du domaine d'étude, augmentant drastiquement N . Pour des simulations turbulentes dont le Reynolds sur l'échelle de Taylor est $Re_\lambda = 90$, la viscosité $\nu = 10^{-3} \text{ m}^2.\text{s}^{-1}$ et une résolution de $N = 128$ sont nécessaires. L'utilisation d'IA se fait à l'aval de cette étape de discrétisation du domaine pour la physique ; l'IA est utilisée comme un module prenant un champ de vitesse en entrée et renvoyant les six composantes différentes de τ_{ij} . En lui proposant tout le volume, on lui demandera de traiter simultanément près de deux millions de points pour chacune des entrées et sorties, et ce nombre peut aisément être multiplié par un facteur 10 voire 100 dès que l'on entend modéliser un écoulement plus turbulent (plus proche des écoulements réelles dont les Reynolds sont incroyablement élevés en général). Cette taille d'entrée et de sortie interfère avec les limitations internes des allocations de mémoire propre à TensorFlow (Abadi et al., 2016) et Keras (Chollet et al., 2015) que nous utilisons. Mais même en outrepassant cette limite, le nombre de simulations nécessaires à effectuer en amont pour assurer des échantillons assez différents et représentatifs des phénomènes que nous recherchons, alourdira davantage l'utilisation de réseau de neurones 3D². Des précautions supplémentaires seront à considérer pour s'assurer de prendre en compte les comportements aux grandes échelles ainsi qu'aux plus pe-

2. De plus, ce manque de rapidité rend encore plus chronophage la sélection des meilleurs hyper-paramètres parfois nécessaire pour atteindre ne serait ce qu'un minimum global.

tites échelles, imposant à priori une profondeur accrue du réseau. Les problèmes d'extinction de gradients - ou moins drastiquement - de propagation de l'information sont à prévoir dans les passes avant et/ou arrière de l'entraînement. Ces considérations ajoutent en fait un nombre colossal de degrés de liberté à explorer pour trouver les couples architectures/hyper-paramètres optimaux ; au bout du compte cela atténue voire empêche l'utilisation des CNN en volume.

Discretisation en volume et apport de Kraichnan - Pour parer à cette limitation, nous proposons de découper les champs d'entrée et de sortie en sous-domaines. Ce découpage se fait en aval de la discrétisation faite pour la résolution des équations de Navier-Stokes. Nous désignons par N_c le nombre de points de discrétisation contenus dans chaque sous-domaine. Dans le cas général, ces discrétisations peuvent varier selon les trois directions de l'espace avec x , y et z , mais dans notre cas les discrétisations sont identiques dans ces trois directions. Les pas de discrétisation sont donc identiques, on note cette valeur commune Δ . Nous découpons alors le volume global en cube dont le côté est mesuré par L_c

$$L_c = N_c \Delta \quad (4.2.1)$$

Ils comprennent N_c^3 points. Comme nous l'avons souligné précédemment, ce découpage en cubes doit prendre en compte la non localité des interactions dans l'espace des modes impliquant le concours de structures aux échelles différentes dans l'évolution temporelle des grandeurs turbulentes. Ceci impose un minimum sur les dimensions des cubes que l'on veut extraire du domaine global. Des tests avec une taille $L_c = 16 \Delta$ se sont avérés très satisfaisants offrant un compromis entre bonnes estimations des composantes du tenseur des contraintes sous mailles et temps de calcul. Nous pouvons établir une relation entre le nombre de Δ par direction et la perte d'information utile à la modélisation des flux de transferts à travers la coupure dans l'espace de Fourier. En effet dans l'étude des transferts énergétiques et l'élaboration des modèles de viscosité turbulente dits *spectraux*, Kraichnan (Kraichnan, 1976) étudie l'évolution de la fraction du transfert total d'énergie à travers la coupure $\Pi(K|K_C)$ en fonction du rapport κ/κ_C , avec κ_C le nombre d'onde de coupure et κ les nombres d'ondes compris dans l'intervalle $[0, \kappa_C]$. La figure issue de cet article est reproduite Fig.(4.2).

La théorie de l'échantillonnage de Shannon permet de lier nombre d'onde maximal et taille de structure minimale correspondant dans l'espace Physique. Nous notons cette taille limite L_{\min} et elle est égale au double du pas de discrétisation, c'est-à-dire

$$L_{\min} = 2 \Delta \quad (4.2.2)$$

Le filtrage des données augmente ce seuil d'un rapport $R = K_{\max}/K_C$; puisque nous considérons des champs de vitesse filtrés conformément aux méthodes LES, c'est cette valeur seuil qui nous intéressera dans la suite. Elle sera désignée par l'appellation de longueur minimale post-filtrage et on l'écrit

$$\tilde{L}_{\min} = R L_{\min} = 2R \Delta \quad (4.2.3)$$

Le choix de $L_c = 16 \Delta$ impose une valeur du rapport κ/κ_C dépendant de R :

$$\kappa/\kappa_C = \tilde{L}_{\min}/L_c \quad (4.2.4)$$

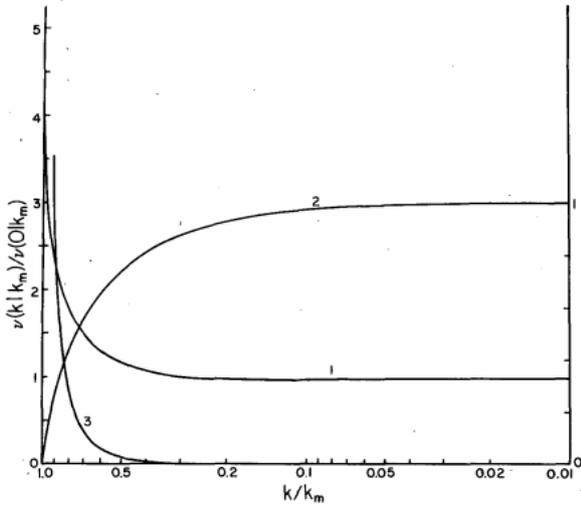


Figure 4.2 – La courbe 2 illustre la relation entre la quantité d’énergie transférée à travers la coupure et les modes associés. La grandeur k_m est le nombre d’onde de coupure considéré dans l’opération de filtrage, que nous avons noté κ_C . Les deux autres courbes dépassent le cadre de ce manuscrit. Figure issue de Kraichnan (1976).

Nous nous plaçons dans le cas où $R = 2$. Comme nous le soulignerons dans la suite, il est primordial de conserver ce rapport. Pour l’heure ce choix impose $\kappa/\kappa_C = 1/4$ permettant de nous situer dans la figure Fig.(4.2).

Pour cette valeur, nous estimons que la dynamique contenue dans les cubes contribue à 92% aux transferts de flux d’énergie à travers la coupure. Nous remarquons également que considérer des cubes de côté $L_c = 8 \Delta$ entraîne une perte de 40% de ces transferts et qu’au contraire, considérer des longueurs plus grandes n’offre pas forcément beaucoup plus d’informations par rapport aux désavantages que cela entraînerait. Dans le tableau Tab.(4.1), nous proposons l’étude de la fraction des transferts capturée par la dynamique dans les cubes pour plusieurs valeurs du rapport κ/κ_C . Cela permet également de considérer l’impact sur l’information conservée des grandeurs N_c et R . Ce tableau illustre bien les limites imposées par le découpage par rapport au filtrage. Si nous considérons un rapport R plus grand, par exemple $R = K_{\max}/K_C = 4$, la longueur minimale des structures post-filtrage sera également plus grande $\tilde{L}_{\min} = 8 \Delta$. Si de plus nous conservons la longueur de cube à $L_c = 16\Delta$, la relation (4.2.4) entraîne $K/K_C = 1/2$. En nous reportant figure Fig.(4.2), nous voyons que la dynamique contenue dans ce cube contribuera à environ 60 % des transferts d’énergie à la coupure limitant la qualité de la modélisation des phénomènes sous mailles. Pour un tel R , il sera nécessaire de prendre en

N_c	R	L_c	\tilde{L}_{\min}	κ/κ_C	Pourcentage des transferts recouverts
16	2	16Δ	4Δ	1/4	~ 90 %
16	4	16Δ	8Δ	1/2	~ 60 %
32	4	32Δ	8Δ	1/4	~ 90 %
8	2	8Δ	4Δ	1/2	~ 60 %

Table 4.1 – Couples $N_c - R$ sur le pourcentage de transferts capturés par la dynamique contenu dans le sous-domaine cubique. Les valeurs de \tilde{L}_{\min} sont calculées utilisant la formule (4.2.3).

compte plus d'échelles, plus de dynamique et donc d'augmenter la tailles des cubes d'un facteur R , assurant 90% de $\Pi(\kappa|\kappa_C)$ au sens de Kraichnan. Cependant, la taille des entrées du réseau de neurones est fixe (notons que l'on peut utiliser le padding pour moduler indirectement la taille des entrées; nous en reparlerons dans les ouvertures en fin de rapport); il est donc impossible de considérer des tailles de cubes différentes de celle avec laquelle il a été entraîné. Nous comprenons donc qu'un CNN ne pourra prédire les composantes sous mailles que pour des valeurs de R fixes selon les simulations considérées.

Dans nos travaux, les cubes seront de dimensions $N_c \times N_c \times N_c$ en terme de points de discrétisation, avec $N_c = 16$ le rapport R est fixé à 2 assurant donc 90% des contributions aux flux d'énergie à la coupure.

Ensembles des entrées et sorties - Pour construire les entrées X , nous découpons alors les trois composantes du champ de vitesse filtré $\bar{u}_x, \bar{u}_y, \bar{u}_z$ en cubes composés de 16 points de discrétisation par direction spatiale. L'ensemble des entrées s'écrit

$$X = \{x \in \mathbb{R}^{3 \times 16 \times 16 \times 16} : x = (\bar{u}_x^{i,j,k}, \bar{u}_y^{i,j,k}, \bar{u}_z^{i,j,k})\} \quad (\text{X})$$

les exposants i, j, k représentent les coordonnées des points contenus dans le cube, ils vont de 0 à $N_c - 1$. De façon analogue, nous construisons l'ensemble des sorties Y considérant cette fois les six composantes distinctes du champ des contraintes sous-mailles :

$$Y = \{y \in \mathbb{R}^{6 \times 16 \times 16 \times 16} : y = (\tau_{xx}^{i,j,k}, \tau_{xy}^{i,j,k}, \tau_{xz}^{i,j,k}, \tau_{yy}^{i,j,k}, \tau_{yz}^{i,j,k}, \tau_{zz}^{i,j,k})\} \quad (\text{Y})$$

Le fait de considérer des données filtrées s'inscrit dans un but sur le plus long terme de conception de modèles LES augmentés par IA. Le filtrage s'effectue selon le rapport $R = K_{max}/K_C = 2$ avec K_C le nombre d'onde de coupure. Ce rapport devra être conservé lors de l'évaluation de l'apprentissage, de la généralisation et des montées en Reynolds chapitre 5.

Après avoir découpé un certain nombre de champs instantanés de la sorte, les paires sont mélangées au sein de l'ensemble d'apprentissage de façon à ce que le réseau extrait les caractéristiques de chaque entrée indépendamment de son origine ou de sa localité dans l'espace. Chaque paire $\{x; y\}_i$ sera considérée indépendante de toutes les autres. Ce découpage en cube permet donc une variété accrue des entrées selon les champs et selon les régions, augmentant d'une part l'expressivité du réseau et *a priori* permettant de réduire le nombre de simulations à effectuer en amont pour construire la base d'apprentissage. Nous y voyons également deux autres avantages majeurs :

- 1- Puisque les cubes sont considérés indépendants l'un de l'autre (même en étant adjacents dans l'espace), ce découpage permet une parallélisation idéale pour l'obtention des composantes du tenseur des contraintes sur tout le domaine de fluide, et notamment dans la prédiction post-entraînement. L'indépendance totale des cubes s'adapte donc très bien aux méthodes de parallélisation et/ou d'hyper-threading en particulier sur carte graphique.
- 2- Du moment que nous vérifions que les rapports $N_x/N_{c_x}, N_y/N_{c_y}$ et N_z/N_{c_z} sont entiers (avec N_x, N_y et N_z le nombre de points de discrétisation de l'ensemble du domaine selon les trois directions de l'espace), nous pouvons envisager des discrétisations du volume suivant les directions de l'anisotropie d'écoulement comme la couche limite ou autre. Les différentes tailles $L_{c_x, y, et z}$ devront être éventuellement adaptées selon les travaux de Kraichnan.

Code pour les simulations - Toutes les simulations dont il est question dans ce manuscrit ont été réalisées avec le code Turbo (Teaca et al., 2009). C'est un code spectral qui base l'évolution temporelle sur un schéma Runge-Kutta d'ordre 4 avec dé-aliasing pour éviter l'impossibilité de distinguer les plans d'ondes (provenant de la discrétisation de l'espace de Fourier) et de générer des erreurs dans l'évolution temporelle de la vitesse ; nous revoyons à la documentation des auteurs pour de plus amples détails sur le code³. Ces simulations sont directes (DNS). Elles sont initialisées par des champs de vitesses aléatoires (voir page 46 de la documentation) ; un terme de forçage est constamment rajouté à l'équation de quantité de mouvement : de l'énergie cinétique est injectée dans les coquilles sphériques de rayon $\kappa \in [2.5, 3.5]$ avec $K_{inj} = 0.5$ identique pour toutes les simulations. Toutes les informations à propos de ce forçage sont disponibles dans la documentation du solver Turbo. De même le CFL est fixé à 0.5. Pour le filtrage, nous utilisons un filtre gaussien défini de façon identique à (Pope, 2001, chap.13) et Zhou et al. (2019) ; nous le définissons par sa fonction e transfert :

$$\widehat{G}(\kappa) = \exp - \frac{\kappa^2 \Delta^2}{24} \quad (4.2.5)$$

L'initialisation des champs de vitesse se fait à partir d'un spectre avec $\alpha = 0.5$ et $\beta = 1.4$ pour l'amplitude et la variance du champ aléatoire généré. À l'initialisation, le spectre du champ de vitesse est représenté figure Fig.(4.3). Nous rappelons enfin que le domaine est cubique de dimension $2\pi \times 2\pi \times 2\pi$, et que les conditions aux limites seront toujours tri-périodiques.

Établissement de l'ensemble d'apprentissage - Nous n'utilisons qu'une seule simulation pour l'établissement des ensembles de données d'apprentissage et de test. Ce choix est motivé par l'hypothèse d'universalité de la Turbulence isotrope selon les hypothèses de Kolmogorov. De plus, loin des effets de bords, il semblerait que la turbulence au sein de chaque écoulement

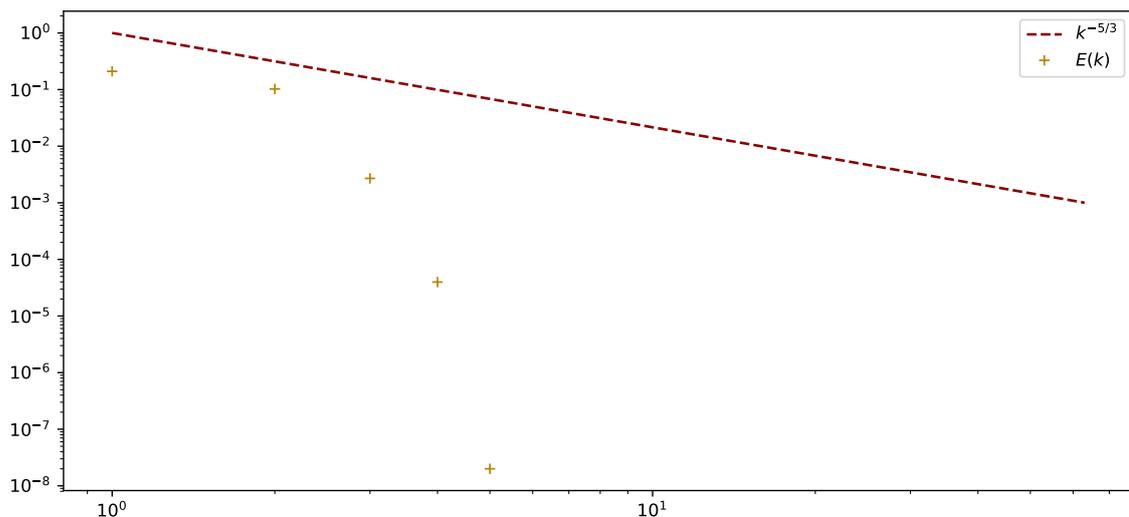


Figure 4.3 – Spectre initial du champ de vitesse

3. Pour plus de détails sur ce solver, voir <https://aqua.ulb.ac.be/home/turbo/>

tende vers l'isotropie et l'homogénéisation due au caractère aléatoire de la turbulence et à la cascade énergétique et structurelle décrits dans le chapitre 2. Nous nous plaçons donc après l'établissement du régime développé de la turbulence⁴. Ce choix nous permettra de discuter des relations entre U_i et τ_{ij} au fur et à mesure que nous monterons en Reynolds. De plus, notre stratégie de discrétisation du domaine en cubes indépendants permettra au réseau de considérer chaque région indépendamment enrichissant son "expérience", l'information prise en compte. La simulation que nous sélectionnons pour construire les ensembles d'apprentissage et de test est effectuée sur un domaine de résolution $N = 128$ et de viscosité cinématique $\nu = 0.001 \text{ m}^2.\text{s}^{-1}$. La simulation est initialisée avec un champ aléatoire. Les évolutions temporelles des grandeurs turbulentes comme l'énergie cinétique ou du Reynolds à l'échelle de Taylor (entre autres) de cette simulation sont disponibles figure Fig.(4.4). Le régime développé est atteint pour l'itération 5000 environ. À cette itération énergie cinétique turbulente, temps de retournement et autres grandeurs oscillent légèrement autour d'une valeur moyenne. C'est à ce stade que nous mesurons le Reynolds à l'échelle de Taylor et c'est avec cette mesure que nous caractérisons l'état de turbulence de la simulation. Le Reynolds Re_λ de la simulation utilisée pour référence est $Re_\lambda = 90$.

À partir de cette itération, nous sélectionnons 10 "instantanées" décorréliées entre elles par plusieurs centaines de temps de retournement. Pour l'ensemble d'apprentissage, ces itérations sont (en centaines) :

$$n_{\text{iterations}}^{\text{train}} = \{50, 55, 60, 65, 70, 75, 80, 85, 90, 95\} \quad (4.2.6)$$

elles sont marquées par des cercles rouges sur la figure Fig.(4.4). La base d'apprentissage est donc composée de $10 \times (N/L)^3$ paires dont les entrées et sorties appartiennent aux ensembles (X) et (Y) respectivement, soit 5120 paires d'éléments au total. La qualité d'apprentissage au cours des époques est évaluée par la fonction de coût ainsi que les métriques définies en amont.

Établissement de l'ensemble de test - L'ensemble test est construit sur les simulations correspondants aux deux premiers cercles jaunes de cette figure, nous écrivons :

$$n_{\text{iterations}}^{\text{test}} = \{22, 44\} \quad (4.2.7)$$

Ces itérations ont été choisies bien loin de celles utilisées pour l'ensemble d'apprentissage dans le but de déterminer si le réseau apprendait une relation structurelle entre entrées et sorties, ou bien s'il apprendait à mimer les données d'apprentissage. La raison pour laquelle nous ne considérons que les premières itérations (et non les autres marquées par les cercles jaunes Fig.(4.4)) est la suivante : les résultats étaient très satisfaisants sur l'ensemble des quatre itérations de test. Or les autres itérations (66 et 88) sont plus proches de celles présentes dans l'ensemble d'apprentissage. Pour surligner davantage la généralisation du réseau, nous avons uniquement considéré les itérations les plus excentrées par rapport à l'apprentissage. Cette façon de procéder pousse l'idée de l'ensemble de test un peu plus loin que ce que l'on considère traditionnellement. Ici aussi, fonction de coût et métriques sont utilisées pour l'évaluation du réseau sur cet ensemble, pour chaque patch de données, à chaque époque de l'apprentissage. L'ensemble de test est composé de 1024 éléments soit 20% de la taille de l'ensemble d'apprentissage.

4. C'est dans cette optique que nous considérons une turbulence forcée et non en décroissance libre.

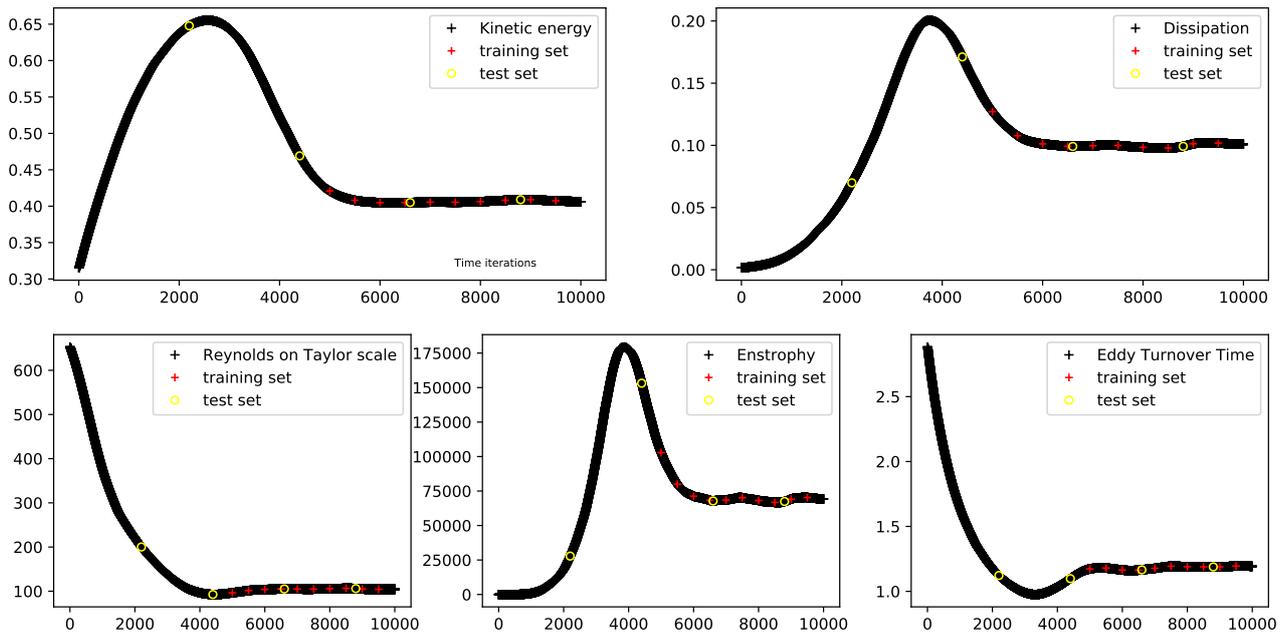


Figure 4.4 – Évolutions temporelles des grandeurs turbulentes (croix noires) dans la simulation utilisée pour la construction des données d’apprentissage. Les croix rouges représentent les itérations des instantanées utilisées pour construire l’ensemble d’apprentissage ; les cercles jaunes représentent celles utilisées pour l’ensemble de test (ou validation).

Nous n’avons pas recours aux méthodes classiques comme la validation croisée (*cross-fold validation*) car l’ensemble de test contient des données radicalement différentes et décorréélées entre elles et de l’ensemble d’apprentissage. Enfin, le sur-apprentissage est évité en considérant une régularisation $L1$ des poids de chaque couche de convolutions, réduisant les degrés de liberté superflus du réseau (voir Fig.(3.6) et les discussions associées section 3.3).

Enfin, pour éviter une saturation du réseau, nous standardisons les données en entrées de sorte que la moyenne de chaque composante soit nulle et de déviation unitaire (Goodfellow et al., 2016). Les moyennes et déviations sont calculées sur l’ensemble des données d’entrée. Les entrées pour le test ou la prédiction seront systématiquement standardisées par rapport aux moments de l’ensemble d’apprentissage.

Architectures - Les travaux utilisant des données en volume sont assez rares dans la communauté des réseaux de neurones et cela ne fait pas exception dans le domaine de la Turbulence. Nous avons alors exploré plusieurs architectures de réseau et très vite le réseau U-Net s’est imposé comme étant le plus prometteur. Dans la section suivante nous détaillons les caractéristiques intrinsèques de cette architecture, que nous identifions comme cruciales.

4.2.2 L’architecture U-net et ses propriétés

Pour illustrer la discussion suivante nous renvoyons le lecteur à la figure présentant l’architecture du réseau à convolutions appelé U-net par ses créateurs (Ronneberger et al., 2015) (à cause de

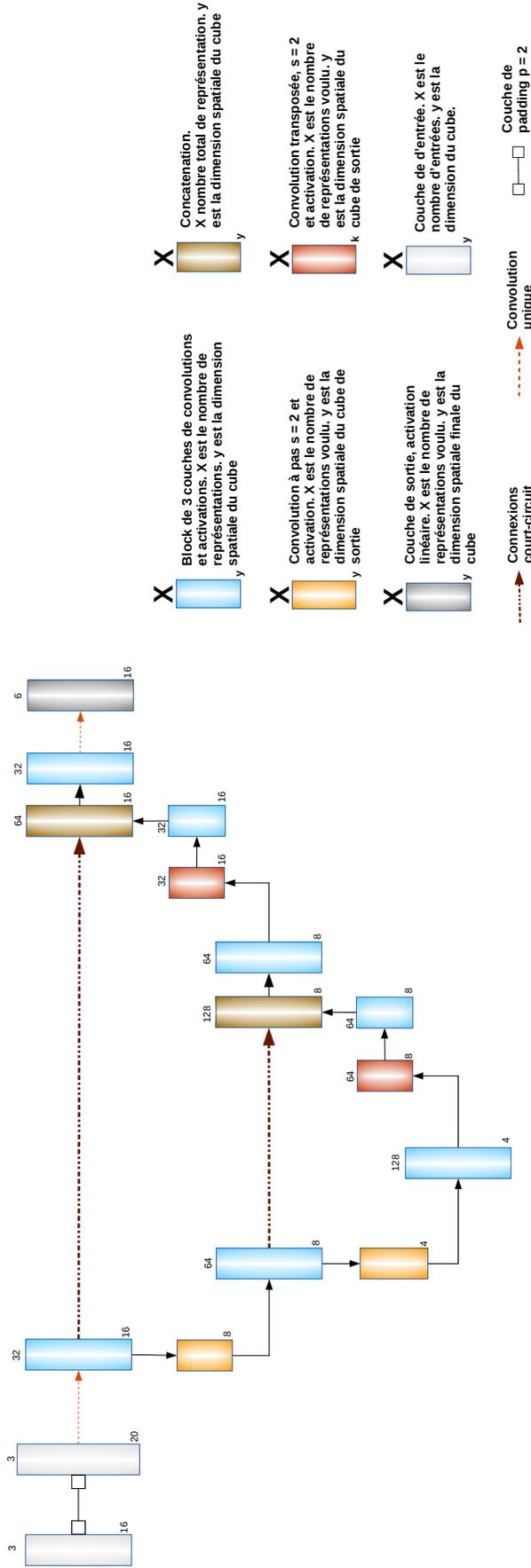


Figure 4.5 – Réseau de neurones à convolutions U-net tel qu'il a été utilisé dans ces travaux. Le nombre de convolutions par étages (représenté par les blocs bleus voir légende (4.5b)) varie dans certains cas. Ce paramètre est appelé *Number of convolutions* voir notamment Tab.(4.2).

sa forme en U) figure Fig.(4.5a) avec sa légende détaillée Fig.(4.5b).

Encodeur Décodeur - Le réseau U-net présente deux étapes majeures dans le traitement de l'information fournie : dans une première phase l'information est contractée ou *encodée*. Cette compression se fait en plusieurs étages. À chaque étage, l'information spatiale est résumée dans des cubes dont la taille (spatiale) est réduite de moitié. Cette opération est effectuée par une convolution dont le pas est $s = 2$ c'est ce que nous avons appelé le pooling non supervisée : les critères de réduction sont définis durant l'apprentissage voir 3.1.2. Ce sous échantillonnage ou *downsampling* est compensé par un nombre de cartes activations doublé à chaque étape de compression ; ceci permet au réseau d'avoir la flexibilité nécessaire pour exprimer les caractéristiques spatiales en patterns bien particuliers et plus complexes. Concrètement les données sont comprimées de deux manières différentes : durant les convolutions chaque opération encode intrinsèquement l'information entrante fournissant une représentation plus abstraite à la couche de convolutions suivante. Le sous échantillonnage en fin d'étage abstrait d'une façon plus intense le signal entrant ainsi que ses caractéristiques et leur cohérence spatiale.

La deuxième phase est une phase extensive appelée *décodeur*. Cette phase de décompression des données se découpe également en paliers. Le sur-échantillonnage ou *upsampling* est réalisé à partir de l'opération de convolution transposée dont le pas est $s = 2$. L'extrapolation est également non supervisée : le réseau apprend à extraire les critères optimaux à cette extension, en cohérence avec l'optimisation globale. À l'issue de cette décompression, le nombre de représentations par convolutions est réduit de moitié conformément au compromis dimensions spatiales/nombre de représentations évoqué dans le paragraphe précédent⁵. Le dernier étage de décompression est directement lié à la couche de sortie du réseau, et la dernière convolution dont la sortie est de taille $n_{\text{batch}} \times 6 \times 16 \times 16 \times 16$ est suivie d'une activation linéaire, pour restituer le signal tel qu'il est obtenu à partir de cette ultime convolution⁶. Rappelons que le nombre n_{batch} correspond au nombre de paires $\{x; y\}_i$ par patch dans l'apprentissage stochastique décrit section 3.1.2.

Encodeur et décodeur sont reliés par le pont (c'est la barre horizontale du "U"). L'information qui transite dans cette section est la version la plus compacte et abstraite des signaux émis en fin de chaque étage de compression. Le U-net permet donc de propager cette information vers les niveaux de décompression "en face" c'est-à-dire dont les dimensions spatiales correspondent avec celles des niveaux de compression. Ces connexions intermédiaires sont désignées sous l'appellation connexion court-circuit ou *shortcut connections* (flèches en pointillés violet dans le schéma Fig.(4.5a)). D'après les auteurs de cette méthode et les études postérieures (voir par exemple Zhang et al. (2018)) ces connexions sont une des forces majeures du U-net⁷.

5. L'abstraction des données opérée par les convolutions est intrinsèque aux neurones.

6. Activer cette convolution à la manière du ReLU ou Mish par exemple, pourrait amener à des aberrations poussant le réseau à compenser la perte d'information par des biais très forts pouvant saturer le réseau et/ou corrompre l'apprentissage

7. **Complément d'information sur l'importance de la non-supervision** : l'architecture convergente/divergente permet au réseau de réinterpréter les données selon des critères définis de façon totalement non supervisés. L'*Information Bottleneck Principle* (Tishby et al., 2000) comprend cette réinterprétation en fonction de l'évolution de l'information mutuelle entre entrée et sortie à travers des couches de neurones au cours de l'apprentissage. Ces études dégagent deux phases dans la compression des données :

- Phase de reconnaissance : le réseau capture les informations utiles pour la prédiction de la sortie imposée et construit les premiers patterns.

Connexions court-circuit - L'architecture du U-net permet différents canaux de propagation de l'information situés à différentes étapes du traitement du signal d'entrée au travers les convolutions. De façon symétrique, le premier palier de compression communique sa version de l'entrée au signal arrivant dans le dernier étage de décompression et ainsi de suite. Le réseau propagera un signal dans lequel les patterns les plus complexes seront combinés (par concaténation) avec ceux les plus superficiels, issus des premières ré-interprétations. Cela permet plusieurs avantages (Ronneberger et al. (2015), Zhang et al. (2018)) :

- Au fur et à mesure que le réseau s'approfondit, les patterns deviennent plus spécialisés. Dans la phase d'extension, il est possible que le réseau n'arrive plus à associer un pattern spécifique à une région de l'espace (par exemple le sourcil au dessus des yeux). Il est alors nécessaire de contextualiser les patterns les plus complexes et les plus déterminants concernant la précision de la prédiction. Nous utilisons alors la concaténation des activations complexes et superficielles et les recombinaisons en appliquant les convolutions sur ce nouveau signal.
- Dans le cadre de réseaux profonds, l'information peut se retrouver coupée et perdue à cause de l'extinction de gradient notamment. Généralement, les zones les plus superficielles du réseau sont les plus touchées par ce problème. Au fur et à mesure, l'optimiseur n'a plus accès à ces zones, l'entraînement se dégrade et devient inefficace. En utilisant ce genre de connexion coupant au travers l'architecture, nous assurons que l'information transite bien de bout à bout du réseau profitant alors au maximum des couches de convolutions successives.

Ce dernier point rappelle l'idée des connexions résiduelles introduites par He et al. (2015) permettant à l'information de "sauter" au dessus de certaines convolutions pour contourner le problème de détérioration propre aux réseaux profonds et très profonds. L'idée est de forcer la connexion entre certains points du réseau et de prédire le résidu $\mathcal{F}(x) = \mathcal{O}(x) - x$ plutôt que la convolution en elle-même $\mathcal{O}(x)$. L'erreur est ensuite compensée par l'ajout de l'information entrante x créant un canal court-circuitant toutes les opérations entre l'entrée x et la sortie $\mathcal{F}(x)$ voir la figure Fig.(4.6).

L'idée des connexions court-circuits s'inspire donc des connexions résiduelles, mais ne la copie pas puisque l'addition est remplacée par la concaténation. Cette concaténation permet donc de conserver le contexte de chaque patterns les plus spécialisés⁸. Dans le cadre des travaux présentés dans ce manuscrit, la profondeur optimale du réseau n'a pas nécessité l'utilisation des connexions résiduelles en plus des connexions court-circuits. En revanche, des tests sans les connexions court-circuits se sont révélés très mauvais, dans lesquels *aucun apprentissage n'était*

- Phase de compression optimale : le réseau comprime l'information sur l'entrée en laissant de côté les caractéristiques de X non informatives sur Y .

Au terme de ces deux étapes, le réseau obtient une représentation minimale et statistiquement suffisante du signal entrée X notée \tilde{X} permettant de conserver au mieux l'information mutuelle $I(X; Y)$. L'architecture particulière de l'U-net autorise cette réinterprétation minimale et suffisante de l'entrée et assure de plus qu'aucune information capitale ne soit perdue au terme des compressions par les convolutions et les réductions/extensions spatiales. Dans la suite du texte, l'utilisation du tilde ne sera pas liée avec la réinterprétation minimale et suffisante.

8. Notons que pour des U-net profonds, des modules de connexions résiduelles sont envisagés (Jha et al. (2019) ou Diakogiannis et al. (2020)), les court-circuits ne se substituent donc pas aux connexions résiduelles dans le cadre de réseaux très profonds

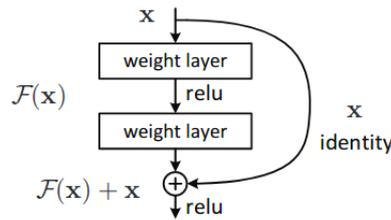


Figure 4.6 – Connexion résiduelle et propagation alternative de l'information He et al. (2015)

possible ou très peu. Cette remarque appuie l'importance des combinaisons d'informations de différents niveaux en ce qui concerne notre objectif. Enfin, notons que l'association de patterns de différents niveaux de complexité encourage le réseau à apprendre des caractéristiques à des échelles spatiales variées. Ces forces font de l'U-net l'un des modèles les plus fiables existant dans le domaine du *deep learning* dès lors que le phénomène visé fait apparaître une grande variété d'échelles spatiales (Reina and Panchumathy, 2018).

De la segmentation d'image à l'expression de gradients - Initialement le U-net fut élaboré pour systématiser la segmentation de représentations visuelles obtenues par les méthodes d'imagerie médicale. Son but était de délimiter cellules, organes etc. au sein de ces images. Pour obtenir cette partition, l'algorithme doit au préalable détecter les délimitations de chaque élément au niveau de ses caractéristiques : couleur, texture et autres. Très performant pour la segmentation, le U-net est intensément utilisé dans le cadre de la recherche Biomédicale (Lassen-Schmidt et al. (2020) ou Taghanaki et al. (2019) entre autres).

Au niveau de son utilisation dans la physique en tant qu'outil de prédiction de champ, les travaux de Lapeyre et al. (2019) sont pionniers et constituent un des rares exemples d'utilisation de données 3D et de CNN de manière générale. Dans le cadre de simulation aux grandes échelles (LES) des phénomènes de flammes turbulentes, la surface de densité de flamme généralisée $\bar{\Sigma} = |\bar{\nabla}c|$, avec c une grandeur caractérisant l'évolution de la réaction (construite comme une variable de progression en fonction de la température ou de la fraction de masse du combustible présent dans le mélange), apparaît comme une inconnue dans l'équation régissant l'évolution temporelle de la concentration $\rho\bar{c}$. Théorie et expériences essaient d'exprimer Σ en fonction de $|\nabla\bar{c}|$ par des approches variées (fractales, analyse multi-fractales etc). Les auteurs proposent de calculer la fonctionnelle $\Sigma \equiv |\bar{\nabla}c| = \mathcal{F}(\bar{c})$ par intelligence artificielle utilisant le U-net et des données 3D comme nous l'avons mentionné. Les résultats obtenus dépassent les formulations algébriques issues de la théorie, démontrant que le U-net est capable de l'opération analogue à la segmentation (tâche de classification) qui est le calcul de dérivées (tâche de régression).

Cette capacité est très importante à nos yeux car la majorité des modélisations de τ^{SGS} font apparaître les gradients des champs de vitesse. Or, ces modèles présentent encore des défauts supposant que l'information du gradient n'est peut être pas suffisante pour construire un modèle sous-maillages. Ainsi, en s'assurant que le réseau peut calculer les gradients si nécessaire, nous pouvons choisir de donner le champ de vitesse filtré total. Dans cette même logique, nous ne fournissons pas le champ de vitesse ET son gradient de peur de brider l'expressivité du réseau et d'éviter toute redondance entre les entrées et les représentations d'une partie des données. Notons enfin, que les cas que nous avons considérés avec les gradients en entrée présentaient des limitations. En effet, comme on le verra dans le chapitre prochain, l'évaluation du modèle

proposé se fera sur deux itérations aux résolutions plus fines et au Reynolds plus élevés. Il s'est avéré que les capacités de généralisation d'un réseau entraîné sur les gradients du champ de vitesse étaient faibles. Nous pensons que les gradients calculés sur une faible résolution n'ont intrinsèquement rien à voir avec ceux évalués sur une résolution plus fine. C'est donc pour toutes ces raisons que le choix du U-net (dont la capacité de calculer des gradients a été établie) s'est imposé.

Après avoir détaillé notre stratégie de construction de bases des données, l'intuition physique guidant le choix du rapport $R = 2$ et du nombre de points N_c pour chaque cube (4.1), nous avons présenté l'U-net au cœur de notre étude, son origine et ses propriétés. Il a été également démontré que ce réseau est capable de traiter des entrées structurées complexes, 3D et d'en calculer les gradients avec une précision dépassant les modèles algébriques (ad-hoc) existants. La dernière étape à couvrir est l'optimisation des hyper-paramètres sur différents niveaux :

- Au niveau des caractéristiques du réseau : nombre de convolutions par étage, taille des filtres, fonction d'activation etc. ;
- Au niveau de l'optimisation : algorithme de minimisation de la fonction de coût, taux d'apprentissage, stratégie et ajustement des constantes de pénalisation etc.

Ces points et d'autres sont abordés dans la section suivante.

4.3 Sélection et impact des hyper-paramètres

L'étude des hyper-paramètres est indispensable tant ces "potards" impactent l'optimalité du réseau. Les valeurs et les choix que nous allons présenter dans cette section sont le résultat de centaines d'essais de combinaisons de paramètres. Parfois, nous avons eu recours aux méthodes de recherche de grille classique ou aléatoire (Goodfellow et al., 2016), parfois nous avons simplement changé un des hyper-paramètres et tenté de comprendre son impact global en figeant les autres paramètres.

Il est difficile d'ordonner tous ces tests pour écrire un résultat complet et lisible. C'est pourquoi nous ne détaillerons pas les explorations effectuées pour l'obtention des valeurs des paramètres évoqués dans la section 4.3.1. Ces valeurs ont malgré tout contribué à l'obtention de résultats globalement meilleurs en terme d'apprentissage et de généralisation. Puis, nous proposons section 4.3.2, une étude succincte sur quelques paramètres qui se sont avérés les plus impactant (parfois innovants) à savoir le choix de la fonction d'activation, la profondeur des étages du U-net et l'algorithme de minimisation. Dans cette section, tous les paramètres de la section 4.3.1 seront considérés comme fixes.

4.3.1 Paramètres fixés en amont

Taille des filtres et padding - Spécifions tout d'abord les dimensions des filtres des convolutions que nous considérons dans le U-net. Le premier étage de compression fait intervenir des filtres de taille $5 \times 5 \times 5$ pour chaque carte d'activation produite. Les convolutions dans le pont font intervenir des filtres de taille $2 \times 2 \times 2$. En dehors, les filtres ont tous une taille commune et standard : $3 \times 3 \times 3$. Une autre particularité qu'il est possible de voir Fig.(4.5a) en amont

du premier étage de compression, est l'ajout de couche de padding dont $N_p = 2$ permettant de considérer plus d'une fois les données situées aux extrémités des bords de chaque cube. Les données sont ensuite re-dimensionnées à l'origine par une convolution dont les filtres sont de dimensions $5 \times 5 \times 5$. Pour profiter de l'auto-réduction des dimensions des données par l'opération de convolution, aucun padding n'est considéré et les dimensions de sorties sont données par la formule Eq.(3.1.7). Dans la suite du réseau, toutes les convolutions comportent des ajouts automatiques de couche(s) et ligne(s) de zéros, de sorte que les dimensions entre l'entrée de la convolution et sa sortie soient identiques (à l'exception des convolutions responsables du sous- et sur- échantillonnage). Le nombre de patch est également fixé à 32.

Différentes phases d'apprentissage - Au niveau des fonctions de coût customisées L_1L_{2M} et L_{1M} , le paramètre Ω responsable de l'impact des contraintes sur l'optimisation générale est fixé à $\Omega = 0.995$. Cela est équivalent à l'utilisation d'une constante de pénalisation $\lambda = 0.005$ sur les contraintes définies Eq.(3.3.3) et Eq.(3.3.4). Cette valeur permet au réseau dans un premier temps de réduire les plus gros écarts entre prédiction et cible (calculés par la distance L_1 , terme en noir sur l'équation Eq.(4.3.1)) accélérant l'apprentissage. Couplée avec les stratégies d'adaptation du taux d'apprentissage et de recherche de direction optimale dans l'espace des paramètres incorporées dans l'algorithme de descente Nadam, la minimisation est très rapide et sans oscillation. Une fois que l'erreur sur la norme L_1 est de même ordre que la pénalisation sur la L_2 (terme en rouge sur cette même équation), le réseau oriente sa recherche de solutions optimales en prenant de plus en plus en compte les contraintes sur les écarts maximaux des événements les plus courants. À ce moment, les recherches adaptatives de taux d'apprentissage réduisent la vitesse d'apprentissage sur les poids les plus expressifs dans la réduction des écarts sur la L_1 et activent des neurones supplémentaires pour rechercher les patterns permettant de prendre en compte la contrainte. Ce faisant, le réseau cristallise l'apprentissage et les performances issues de la première phase de minimisation. Durant tout ce processus, il est primordial de ne pas sur-apprendre et de manière plus générale, ne pas perdre de vue l'objectif l'optimisation. Pour s'en assurer, nous contraignons la flexibilité du réseau par une pénalisation Lasso sur les filtres⁹ de convolutions. Ici aussi, il est impératif de tempérer l'impact de cette nouvelle contrainte pour ne pas enrayer l'apprentissage. La valeur de pondération de cette dernière la plus optimale testée est $\lambda_{L_1} = 1 \cdot 10^{-6}$ (cette pénalisation Lasso est représentée par le terme en bleu équation Eq.(4.3.1)). Il s'est avéré qu'au delà (en terme de facteur 10), la régularisation détériore l'apprentissage; en deçà, elle n'est pas assez efficace et le réseau sur-apprend. La fonction de coût prenant en compte ces deux contraintes s'écrit (pour la L_1L_{2M}) :

$$\mathcal{J} = \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_1 + (1 - \Omega) \max \|Y_{\text{True}} - Y_{\text{Pred}}(\theta)\|_2 + \lambda_{L_1} \|W\|_{L_1} \quad (4.3.1)$$

avec W les poids constituant les filtres de convolution.

Valeur du moment et réduction du taux d'apprentissage - De façon analogue, l'impact du moment classique ou Nesterov a été appuyé par un paramètre constant $\mu = 0.95$ là où la littérature recommande une valeur de 0.9. Il s'est avéré qu'augmenter cette valeur accélère

9. Par convention, les biais ne sont pas pénalisés. On peut trouver un début d'explication de cette convention dans la discussion sur le problème du *dying ReLU* section 3.2.

systématiquement l'apprentissage sans le déstabiliser ou le corrompre et permet de réduire davantage fonction de coût et métriques. Enfin, il est très important de réduire le taux d'apprentissage en cours du processus, y compris dans les méthodes adaptatives comme Adam ou Nadam. De ce fait nous forçons le réseau à chercher des caractéristiques de plus en plus complexes et profondes permettant d'entériner ce qui a été appris et de fournir un entraînement plus général parfois même plus robuste (Goodfellow et al., 2016). Il faut néanmoins noter que cette réduction peut stopper l'apprentissage autant que l'accélérer, nous soulignons l'importance de tester au préalable plusieurs ensembles de valeurs concernant

- La valeur initiale du taux d'apprentissage. La valeur la plus optimale que nous avons testée est $\eta_0 = 2 \cdot 10^{-4}$. Au delà, l'algorithme Nadam explose presque systématiquement. Ceci est peut être causé par l'utilisation du NAG qui théoriquement devrait être initialisé par une formule le reliant avec μ ceci n'est pas certain car Sutskever (2013) suppose qu'une initialisation à la main est plus propice d'être fonctionnelle (chapitre 7 p.82).
- La différence minimale d'amélioration en deçà de laquelle le taux d'apprentissage doit être diminué. Cette tolérance a été fixée à $5 \cdot 10^{-4}$. La valeur minimale du taux d'apprentissage en dessous de laquelle il ne sera plus modifié a été fixée à $\eta_0/5$.
- La patience avant une telle diminution est exigée. Dans notre cas, elle a été fixée à 150 époques pour éviter que le taux d'apprentissage ne soit trop bas trop tôt.
- Le schéma de réduction : en escalier, exponentielle etc ; ainsi que le facteur de déclin. Nous utilisons une réduction en étage avec un facteur de 0.9 dès lors que les conditions discutées au dessus sont vérifiées. Enfin, nous imposons un seuil sous lequel cette réduction n'est plus appliquée.

Ces hyper-paramètres ne seront plus modifiés dans la suite.

4.3.2 Illustration de l'impact de certains hyper-paramètres

Nous avons souvent souligné l'importance des hyper-paramètres dans ce manuscrit. Pour des raisons évidentes, nous ne pouvons pas présenter les centaines voire milliers de tests que nous avons effectués pour parvenir au modèle optimal au cœur du chapitre 5. Nous discutons plutôt des hyper-paramètres dont l'impact est le plus fort : les fonctions d'activation et des influences de leurs propriétés, la profondeur du réseau et la capacité d'apprentissage, et enfin les algorithmes de minimisation de la fonction de coût et la différence entre moment classique et Nesterov. Notons que les tests que nous présentons sont pour la plupart bons puisque les autres hyper-paramètres ont été optimisés.

Fonction d'activation : saturation et performance - Parmi les propriétés les plus remarquables des fonctions d'activation discutées section 3.2, le ratio saturation/creusalité de l'activation semble être fondamental pour un apprentissage optimal. Dans un problème de classification, la saturation aide la catégorisation des entrées favorisant la prise de décision. Les fonctions d'activation dites saturantes ou double saturantes comme Tanh et Sigmoides sont donc très populaires dans ce genre de travaux. Pour une tâche de régression, que l'on peut voir comme une classification dans laquelle les "classes" sont infinies, la saturation apparaît rarement comme la meilleure solution. Les nombreux articles mentionnés section 4.1 utilisant ces fonctions saturantes rajoutent souvent une couche de renormalisation avant chaque convolution

ou couche dense. Ces couches réduisent l'effet de la saturation en confinant les entrées dans la région linéaire de chacune de ces fonctions. La renormalisation est non supervisée et ramène à zéro la moyenne de chaque patch de données entrantes (de chaque couche donc) et réduit l'écart à la moyenne à 1. Les renormalisations sont très efficaces et très utilisées mais elles ajoutent de la complexité à l'apprentissage et peuvent parfois amener à des performances sous-optimales. La fonction ReLU se démarque par une saturation à 0 des entrées négatives. C'est ce que l'on a appelé la creusalité de la sortie ou *sparsity*. Elles agissent comme une régularisation sur les poids du réseau forçant le réseau à construire les patterns les plus efficaces dans le processus plus global de la prédiction. La renormalisation est en général indispensable pour éviter que cette propriété ne devienne son talon d'Achille. Le problème du *dying ReLU* est également un défaut majeur de cette activation inhérente à sa construction. Les fonctions d'activation inspirées par le ReLU (Leaky-ReLU ou PReLU) corrigent et généralisent la fonction ReLU (voir section 3.2). Il est alors possible d'utiliser ces fonctions d'activation sans forcément renormaliser les entrées ou craindre une extinction de tous les neurones. Elles sont moins utilisées dans la littérature. Les fonctions plus récentes Swish et Mish s'inspirent des unités rectificatrices et des fonctions saturantes, elles sont d'ailleurs construites à partir de ces dernières (voir les formules (3.2.6) et (3.2.7)).

Après ce bref rappel, nous voulons désormais tester l'apprentissage des réseaux dont les activations sont à tour à tour les différentes fonctions mentionnées. L'ensemble de l'apprentissage est le même, quoi que mélangé aléatoirement. Les initialisations des poids et des biais ainsi que la profondeur du réseau sont également identiques. Après 2000 époques globales d'apprentissage, nous renseignons dans le tableau Tab.(4.2) les valeurs de la fonction de coût L_1L_{2M} définies par la formule (4.3.1) et les métriques L_1 Eq.(3.3.1) et L_2 Eq.(3.3.2). Les valeurs **en gras** appuient les meilleures performances. On voit alors que les fonctions à double saturations sont bien moins performantes (sans renormalisation) que les autres fonctions d'activation. Les unités rectificatrices sont aussi très performantes au niveau de l'ensemble de l'apprentissage; les fonctions

Activations	L_1L_{2M} (10^{-3})	MAE (10^{-3})	MSE (10^{-6})	VAL MAE (10^{-3})	VAL-MSE (10^{-6})
LeakyReLU ($\alpha = 0.2$)	2.3	1.7	11	2.3	20
Sigmoid	12	10	380	10	670
Tanh	4.9	3.9	46	5.0	82
ReLU	2.3	1.8	11	2.4	22
Mish	2.2	1.5	8.9	2.0	17
Swish	2.3	1.5	9.1	2.0	17
PReLU	2.2	1.7	10	2.4	21

Table 4.2 – Comparatif des performances sur les ensembles d'apprentissage et de test pour des réseaux avec chacun une fonction d'activation différente. La profondeur du réseau est de $NOC = 2$. La police **appuyée** souligne les meilleurs performances. Le suffixe VAL désigne l'ensemble de test.

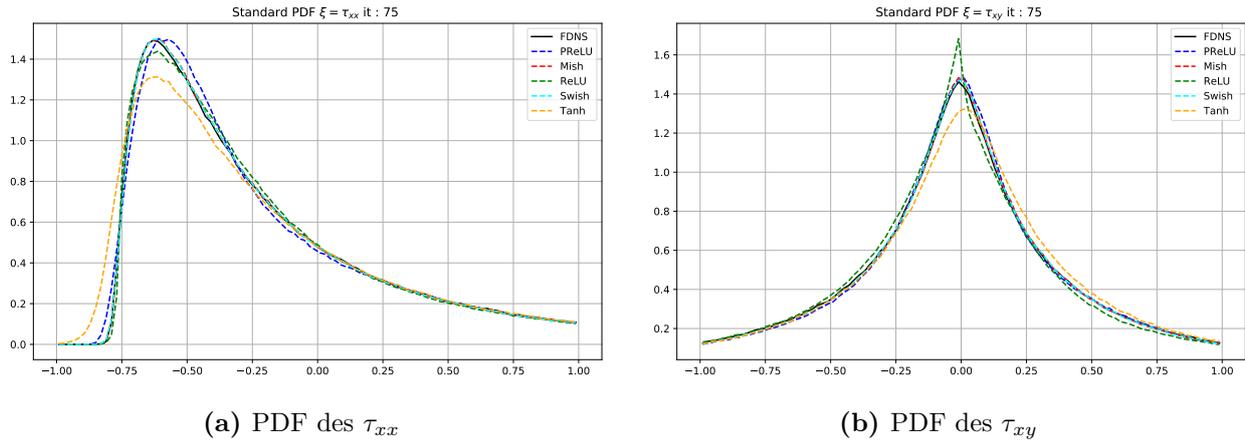


Figure 4.7 – Comparaisons des PDFs des prédictions issues des CNN qui diffèrent entre eux de par leur fonction d’activation. Les différentes lignes brisées représentent ces résultats, la ligne noire constitue la distribution de τ_{xx} et τ_{xy} théorique respective.

Swish et Mish sont légèrement plus performantes que ces fonctions sur l’ensemble d’apprentissage et se détachent au niveau de la généralisation sur l’ensemble de test. On constate également que pour une valeur identique de la fonction de coût à savoir $L_1 L_{2M} = 2.3 \cdot 10^{-3}$ correspondent plusieurs valeurs de métriques (voir pour ReLU et Swish dans ce tableau), nous supposons que la pénalisation Lasso est la cause de ces différences.

Nous visualisons figures Fig.(4.7) l’impact de ces différences sur les fonctions Tanh (ligne brisée orange), ReLU (ligne brisée verte), PReLU (ligne brisée bleue), Mish (ligne brisée rouge) et Swish (ligne brisée cyan) au niveau des distributions (PDF) obtenues à partir de la prédiction de chacun de ces modèles. Prenons par exemple les composantes τ_{xx} (diagonale) et τ_{xy} (extra-diagonale) calculées à partir du même champ de vitesse instantané issue de la simulation de référence à l’itération 75. Nous comparons le tout avec les distributions cibles que constituent les composantes τ_{xx}^{FDNS} et τ_{xy}^{FDNS} (courbes noires). Ce cas fait partie de l’ensemble d’apprentissage. Les quantités prédites par le réseau dont la fonction d’activation est Mish ou Swish ont presque exactement les mêmes valeurs. Les événements rares situés aux extrémités de l’axe des abscisses sont bien retrouvés au même titre que les événements plus courants. Le constat n’est pas le même pour la fonction à double saturation Tanh ; notons que les résultats du réseau utilisant la sigmoïde sont bien moins bons comme on pouvait s’y attendre au vue des performances en fin d’entraînement (voir tableau Tab.(4.2) entrée Sigmoïde). Notons que globalement la composante diagonale véhicule plus d’énergie et présente une gamme d’amplitudes plus grande. Ainsi, les unités rectificatrices semblent également très performantes en particulier le CNN activé par le PReLU ; mais les différences de distributions au niveau de τ_{xx} montrent que le CNN n’a pas totalement intégré les différentes amplitudes, et cela pourrait s’avérer compromettant sur le moyen-long terme. Celui activé par la ReLU émet globalement des composantes moins précises que le précédent réseau. Pour conclure, les fonctions exotiques Swish et Mish permettent une meilleure réponse globale du réseau et semblent parfaitement intégrer les différentes amplitudes présentes dans les composantes diagonales du tenseur. Au vue de leur construction, de leurs objectifs et de leurs propriétés, nous interprétons cette supériorité par le fait qu’elles permettent une exploration de l’espace des solutions plus approfondie en conservant notamment l’information négative (plus ou moins) proche de zéro, tout en ramenant à zéro les entrées trop négatives

Activations	L_1L_{2M} (10^{-3})	MAE (10^{-3})	MSE (10^{-6})	VAL MAE (10^{-3})	VAL-MSE (10^{-6})
LeakyReLU ($\alpha = 0.2$)	2.2	1.5	8.8	2.1	18
Sigmoid	12	10	380	13	700
Tanh	5.2	4.1	53	5.4	94
ReLU	2.3	1.6	9.5	2.2	19
Mish	2.2	1.3	7.7	1.8	14
Swish	2.1	1.3	7.8	1.8	15
PReLU	2.1	1.6	8.6	2.2	18

Table 4.3 – Comparatif des performances sur les ensembles d'apprentissage et de test pour des réseaux qui diffèrent de par leur fonction d'activation. La profondeur du réseau est de $\text{NOC} = 3$. La police **appuyée** souligne les meilleures performances. Le suffixe VAL désigne l'ensemble de test.

profitant ainsi de fluidifier la circulation de l'information au sein du réseau (c'est le principe de la creusalité). Cette supériorité se fait aussi bien au niveau de la L_1 (ce qui correspond aux événements les plus énergétiques) que d'un point de vue des moindres carrés (événements plus courant, oubliant les extrêmes), sans pour autant avoir eu besoin de pénaliser davantage ou de normaliser les entrées.

Seuil du nombre de convolutions par étage - On retrouve d'ailleurs cette supériorité en considérant un U-net plus profond, c'est-à-dire dont le nombre de convolutions par étage de compression/décompression est plus élevé, voir le tableau Tab.(4.3). D'après ce dernier, les performances sont globalement meilleures que précédemment, montrant l'impact de la profondeur du réseau sur les capacités d'apprentissage et de généralisation. Ici encore les fonctions Swish et Mish se démarquent des fonctions à double saturation ou des unités rectificatrices. Notons que les performances saturent à cette profondeur et que rajouter des couches de convolutions ($\text{Noc} > 4$) a plutôt tendance à dégrader l'apprentissage et la généralisation, en plus d'exiger plus de temps de calculs. Utiliser des connexions résiduelles inspirées du ResNet (He et al., 2016) ne s'est pas non plus avéré plus intéressant.

Pour la suite, nous choisissons de conserver le réseau CNN dont la fonction d'activation est Mish et dont la profondeur est $\text{Noc} = 3$. Ce choix, en plus d'être motivé par les meilleures performances, est plus économique en terme de temps de calcul que le CNN de profondeur $\text{Noc} = 4$ et plus précis que le réseau avec $\text{Noc} = 2$. Nous sélectionnons le réseau entraîné avec la fonction Mish plutôt que Swish par rapport à la légère avance de généralisation sur la L_2 du Mish (voir tableau (4.3)).

Algorithme d'optimisation - Un des résultats les plus surprenants de nos travaux est l'incapacité de l'algorithme Adam à optimiser le CNN sur le problème considéré. Nous avons parcouru différentes gammes de taux d'apprentissage par recherche aléatoire (Goodfellow et al.,

Fonctions de coût	L_1L_{2M}	L_{1M}	MAE	MSE	VAL-MAE	VAL-MSE
L_1L_{2M}	$2.2 \cdot 10^{-3}$	x	$1.3 \cdot 10^{-3}$	$7.7 \cdot 10^{-6}$	$1.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-5}$
L_{1M}	x	$2.6 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$	$7.6 \cdot 10^{-6}$	$1.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-5}$
MAE	x	x	$2.0 \cdot 10^{-3}$	$8.46 \cdot 10^{-6}$	$2.5 \cdot 10^{-3}$	$1.6 \cdot 10^{-5}$
MSE	x	x	$9.0 \cdot 10^{-3}$	$2.68 \cdot 10^{-4}$	$1.1 \cdot 10^{-2}$	$4.1 \cdot 10^{-4}$

Table 4.4 – Comparaisons des performances de CNN entraînés avec différentes fonctions de coût.

2016), nous avons également essayé des stratégies d’initialisation différentes, des profondeurs de réseau différentes ; cependant l’algorithme stagne dès les premières étapes de la minimisation (y compris pour les fonctions de coût plus classique telle que la MAE). Prenons par exemple le cas optimal $Noc = 3$ et la fonction d’activation Mish. Le taux d’apprentissage initial le plus probant est $\eta_0 = 0.007$ pour une optimisation par Adam. À la fin des 2000 époques d’apprentissage globales, les valeurs sur les métriques sont de l’ordre de 0.009 pour la L_1 et de $4 \cdot 10^{-4}$ pour la L_2 ce qui est bien loin des résultats obtenus dans le cas de l’optimisation par Nadam tableaux Tab.(4.2) et (4.3).

Retour sur la fonction de coût - Dans les résultats précédents, nous avons supposé que la fonction customisée L_1L_{2M} était la meilleure mais sans pour autant fournir des preuves. Nous proposons de le faire a posteriori en continuant de comparer les métriques MAE et MSE moins abstraites. Pour chacun des cas du tableau Tab.(4.4), les hyper-paramètres des différents CNN sont les mêmes ; nous ne faisons varier uniquement les fonctions de coût d’un réseau à l’autre. Chaque ligne représente les valeurs des fonctions de coût et métriques (colonnes) évaluées à la fin des 2000 époques globales d’apprentissage d’un CNN entraîné sur la fonction de coût donnée en entrée de la ligne (la colonne la plus à gauche). Sur les métriques, on voit une différence nette entre les fonctions de coût classiques et celles customisées. Nous pouvons expliquer ces différences en fonction de ce que nous avons développé précédemment. Le fait que le réseau entraîné sur $\mathcal{J} = \text{MSE}$ fournisse les pires résultats exprime le fait que les carrés sont trop faibles corrompant l’apprentissage et empêchant de toute façon le signal de se propager. Nous comprenons qu’il faut considérer la norme L_1 comme base d’optimisation. En soi, la MAE fournit des résultats qui sont bons, mais pas suffisamment notamment sur l’ensemble de test. Plus globalement, la généralisation qui fait apparaître des erreurs globales est un signe qu’il faut magnifier les erreurs plus faibles et pas forcément prises en compte. C’est l’objet des fonctions de coût et de l’optimisation du paramètre Ω . La L_{1M} et la L_1L_{2M} diffèrent du type d’erreurs considérées. On considère d’ailleurs l’erreur maximale observée, pondérée par le facteur Ω pour permettre avant tout d’atteindre les performances de la MAE, puis de les raffiner. Le maximum est utilisé pour magnifier les erreurs subsistantes, susceptibles de gêner à la généralisation. Cette pénalisation impose des patterns plus précis que ceux obtenus par la L_1 ; nous pensons que la pénalisation impacte surtout les patterns les plus profonds car en général les couches superficielles détournent les caractéristiques les plus globales et générales.

4.4 Évaluation du modèle sélectionné sur les ensembles d'apprentissage et de test

Avant toute chose, nous traçons l'évolution des fonctions de coût et métriques au cours des 2000 époques globales d'apprentissage figure Fig.(4.8). Cette figure permet de voir en simultanément la réduction de la fonction de coût et des deux métriques qui la composent. Les billes représentent les évaluations de la fonction de coût (en rouge), de la MAE (en bleu) et de la MSE (en orange) sur l'ensemble d'apprentissage. Les triangles renversés indiquent la valeur de ces mêmes grandeurs sur l'ensemble de test. Le code couleur est le même à quelques teintes près. On distingue deux temps généraux dans cet apprentissage, on va voir qu'on peut en dégager un troisième intermédiaire entre les deux temps les plus flagrants. Dans un premier temps, on remarque que les trois fonctions diminuent de façon abrupte que ce soit pour les données d'entraînement ou celles de test. En terme d'époques globales, cette phase dure de façon claire jusqu'à l'époque 250. À ce moment, le réseau a vu en fait quarante mille patches de données de taille 32 tirés aléatoirement dans l'ensemble de données d'apprentissage¹⁰; on comprend ici la terminologie stochastique pour désigner la descente de gradient. Un deuxième temps apparaît clairement où les courbes ont une pente assez faible. Cette faible descente marque en fait l'apprentissage des patterns les plus profonds ceux qui permettent des prédictions précises presque au détail près. Le troisième temps intermédiaire est présent de façon significative dans les courbes oranges de la MSE. On note qu'entre les époques globales 250 et à peu près 1000 la pente reste significative jusqu'à ce que l'évaluation sur l'ensemble d'apprentissage soit inférieure au cent-millième. À partir de là, la MSE continue de diminuer jusqu'à atteindre la valeur renseignée tableau Tab.(4.3) (idem pour les autres courbes). Les fluctuations visibles reflètent le caractère stochastique de l'apprentissage. Il est important que l'algorithme d'apprentissage soit robuste à ces pics car sinon l'apprentissage peut être complètement corrompu. Nous pensons que c'est la raison pour laquelle le taux d'apprentissage a dû être nécessairement initialisé à une valeur inférieure à $2 \cdot 10^{-4}$. Enfin, notons que les fluctuations sont quasiment inexistantes

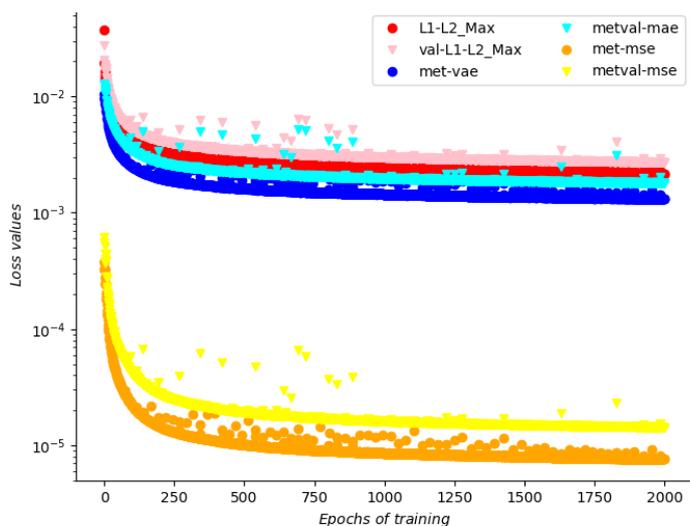
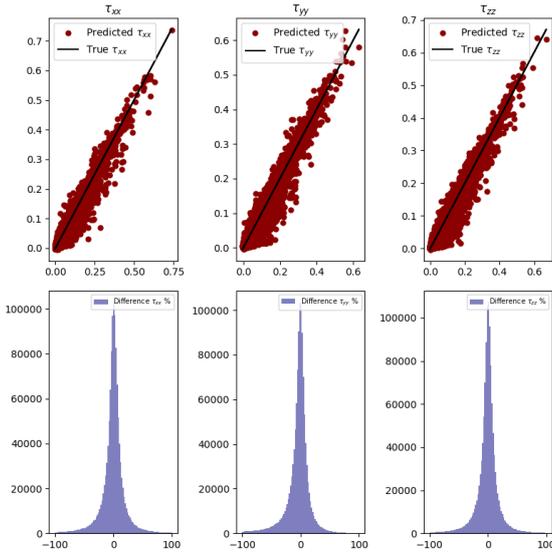


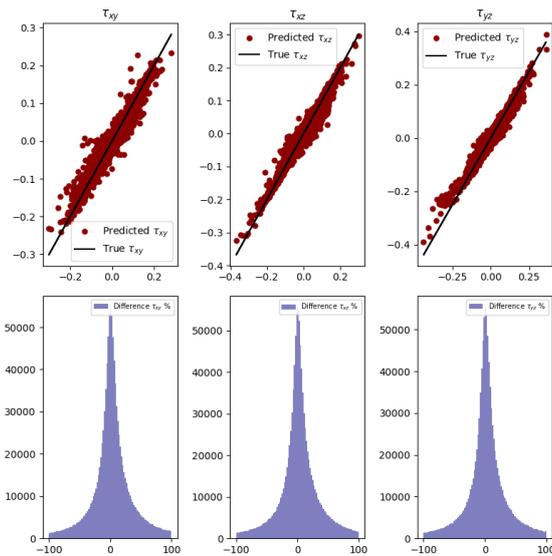
Figure 4.8 – Évolutions des fonctions de coût et métriques en fonction des époques globales d'apprentissage. Les cercles représentent des évaluations de ces fonctions sur l'ensemble d'apprentissage alors que les triangles inversés traduisent leurs scores sur l'ensemble de test

10. Les données tracées dans la figure Fig.(4.8) représentent l'évaluation des métriques sur *la totalité* des ensembles.

vers les dernières époques d'apprentissage, signe que les instabilités dues à la stochasticité de l'entraînement ont été dissipées et que l'apprentissage s'approche du sur-apprentissage. Le fait que les courbes traitant de l'ensemble de test ne s'emballent pas est un signe qu'il n'y a pas d'overfitting. Les deux temps discutés plus haut sont également lié à l'obtention de la réinterprétation minimale et suffisante des données discutée note 7. La dernière étape avant l'utilisation d'un modèle d'intelligence artificielle est d'évaluer ses capacités d'apprentissage et de généralisation. Un graphe très utilisé pour tester l'apprentissage est le diagramme de dispersion. Dans le paragraphe suivant nous nous référons aux figures Figs.(4.9).



(a) Diagramme de dispersion pour τ_{ii}^{ML} .



(b) Diagramme de dispersion pour τ_{ij}^{ML} $i \neq j$

Figure 4.9 – Diagramme de dispersion pour τ_{ij}^{ML}

Diagramme de dispersion et apprentissage -

Nous tirons au hasard (sans remise) 300 événements appartenant à \mathbf{X} ; l'ensemble résultant est noté X_{300}^{Train} . Les 300 cubes considérés contiennent un total de $1,536 \cdot 10^6$ points. Avant de fournir cet ensemble à l'IA, il faut centrer et réduire ces entrées avec les mêmes moyenne et déviation qu'utilisées durant l'apprentissage. On effectue la standardisation suivante :

$$\hat{X}_{300}^{Train} = \frac{X_{300}^{Train} - \mu^{Train}}{\sigma^{Train}} \quad (4.4.1)$$

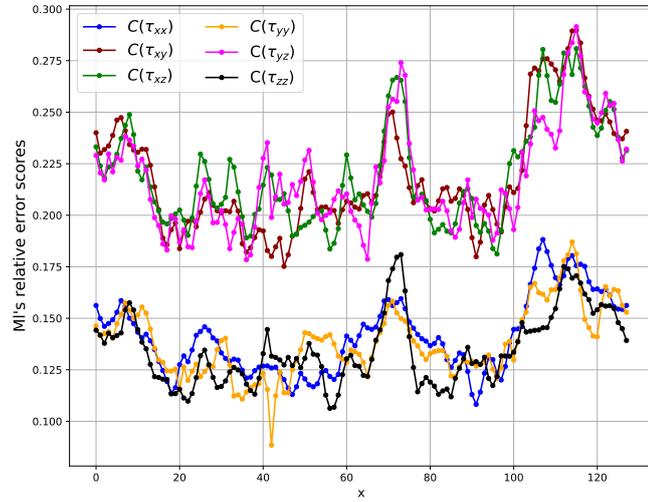
avec μ^{Train} la moyenne sur chacun des points du cube pour toutes les composantes des entrées, calculée sur l'ensemble des données de l'apprentissage et σ^{Train} la déviation de standard équivalente; \hat{X}_{300}^{Train} est donc l'ensemble des entrées recentrées et réduites que nous considérons pour évaluer l'apprentissage.

La prédiction des 300 différentes entrées forme l'ensemble noté \hat{Y}_{300}^{Train} , les quantités exactes, connues, constituent l'ensemble noté Y_{300}^{Train} . Nous traçons la droite $Y = Y$ puis les points $Y = \hat{Y}$. La répartition de ces points autour de la droite des données cibles permet d'avoir un aperçu globale de l'erreur commise lors de la prédiction. Elle permet également d'identifier les gammes de valeurs pour lesquelles les prédictions sont bonnes et éventuellement le contraire. Comme nous évaluons la prédiction des six composantes du tenseur des contraintes, il est nécessaire d'effectuer ces tracés pour chacune de ces composantes. À chacune des composantes, nous faisons correspondre l'histogramme des erreurs absolues entre prédictions et cibles en pourcentage. La superposition parfaite des prédictions avec les données cibles n'est pas recherchée car elle traduirait un sur-apprentissage et donc une faible capacité à généraliser ce qui a été appris. De manière générale, nous pouvons affirmer que les prédictions sont assez

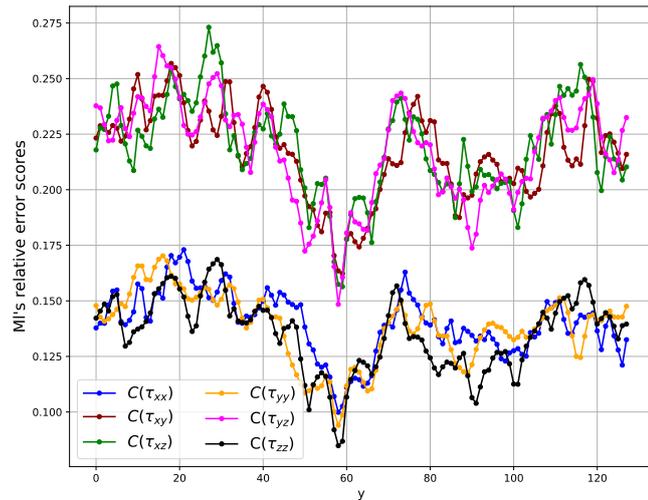
proches des données cibles, même si les erreurs restent conséquentes sur certains points.

À présent nous cherchons à déterminer si une direction est privilégiée dans l'apprentissage où si au contraire, l'apprentissage s'est fait de manière isotrope au sein des cubes. Pour se faire, nous reprenons les champs de vitesse de la simulation de référence à l'itération 75. Nous parcourons les composantes de vitesse en cubes et pour chacun des triplets $x \in X$ recentrés selon la formule (4.4.1), nous effectuons une prédiction des éléments correspondants $\tilde{y} \in \tilde{Y}$. Ce faisant, nous construisons les τ_{ij}^{ML} cube après cube, jusqu'à obtenir le tenseur complet. Nous calculons ensuite l'erreur relative E_r (2.4.14) entre cible et prédiction, le long des trois axes de l'espace. Si les comportements et les amplitudes d'erreurs sont similaires pour chaque graphe, c'est qu'il y a isotropie de l'apprentissage. Les figures Fig.(4.10) représentent les tracés de ce qui vient d'être décrit. Elles font apparaître plusieurs points : tout d'abord, les erreurs commises sont sensiblement les mêmes dans l'espace, l'apprentissage n'a préféré aucune des directions au sein des cubes ; il y a donc isotropie dans l'apprentissage. Deuxièmement, nous confirmons la meilleure estimation relative des composantes diagonales (courbes bleue, orange et noire), par rapport aux composantes extra-diagonales (courbes rouge foncée, magenta et verte). L'isotropie ainsi mise en évidence n'a été encouragée d'aucune sorte pendant l'apprentissage ; de même aucune des composantes n'a été promue pendant l'entraînement ce qui est très encourageant pour la prédiction de toutes les composantes de tenseurs turbulents en volume. Nous renseignons également les scores globaux dans le tableau Tab.(4.5). Pour rappel les scores globaux sont les corrélations Pearson, l'erreur relative telles que présentées dans les équations Eq.(2.4.8) et (2.4.13) ainsi que le score \mathcal{R}^2 utilisé pour mesurer la variance manquante dans les données prédites comparés aux données attendues. On renseignera alors la grandeur $1 - \mathcal{R}^2$. Dans le tableau, nous calculons ces mêmes scores en considérant également τ_{ij}^{Clark} ; ce sont les valeurs en orange. À cette itération issue de l'ensemble d'apprentissage, les prédictions sont plus précises que la modélisation de Clark.

(a) Erreurs relatives le long de l'axe x



(b) Erreurs relatives le long de l'axe y



Les champs sont plus corrélés, expriment plus d'informations aux second ordre assurant une distribution de valeur plus cohérente avec les données FDNS et enregistrent des erreurs relatives plus basses sur chaque composante. Ces résultats concernent évidemment l'ensemble d'apprentissage et sans évaluations similaires pour des données inédites, il n'est pas encore possible de statuer si le défi de construire une IA capable d'effectuer des prédictions volumiques de la totalité du tenseur τ_{ij}^{ML} est relevé. De même pour la supériorité de la prédiction par rapport à la modélisation.

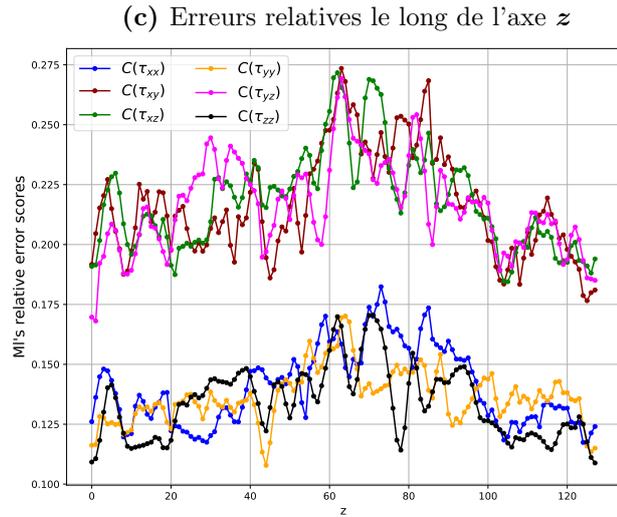


Figure 4.10 – Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales ; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales.

Avant d'évaluer les capacités de généralisation, nous signalons que les études sur les niveaux vectoriel et scalaire dont il était question dans la section 2.4.2 n'ont pas été effectuées ici car nous testons l'apprentissage en soi (le niveau tensoriel) et non son impact sur les grandeurs dérivées. Cette étude sera systématiquement faite dans le chapitre suivant dans lequel nous abordons la question de la faisabilité d'un modèle entièrement basé sur l'intelligence artificielle.

Évaluer la généralisation - Les différentes figures présentées jusqu'ici concernaient des points ou une simulation appartenant à l'ensemble d'apprentissage. Nous voulons à présent tester si le réseau peut généraliser pour des points et des itérations inédites, tout en conservant ce que nous avons appelé l'isotropie de la prédiction en plus de bons scores. Nous procédons de la même manière que précédemment. Nous construisons l'ensemble X_{300}^{Test} en tirant au hasard 300 points dans l'ensemble de test. Après la remise à l'échelle à partir des paramètres de l'apprentissage conformément à l'équation Eq.(4.4.1), ces entrées sont fournies au réseau. Les réponses

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.98 /	0.98 /	0.98 /	0.99 /	0.98 /	0.99 /
	0.97	0.97	0.97	0.97	0.97	0.97
$1 - \mathcal{R}^2$	0.03 /	0.04 /	0.05 /	0.03 /	0.05 /	0.03 /
	0.12	0.07	0.07	0.11	0.07	0.12
E_r	0.14 /	0.21 /	0.22 /	0.14 /	0.21 /	0.13 /
	0.27	0.26	0.26	0.27	0.26	0.27

Table 4.5 – À l'itération 75, Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij} .

obtenues constituent l'ensemble $\tilde{Y}_{300}^{\text{Test}}$, nous traçons Figs.(4.12), les figures équivalentes aux figures Figs.(4.9). La répartition de l'erreur sur ces points jamais vus par le réseau ressemble beaucoup aux résultats obtenus sur l'ensemble d'apprentissage. Les ordres d'erreurs sont sensiblement les mêmes. Les prédictions sont faites sur des cubes de champ de vitesse issus des itérations inédites de la simulation référence. Les bonnes performances sur chacun des points de ces cubes (points symbolisés par les points rouges foncés) doivent être confirmées par le calcul des scores globaux sur l'ensemble du champ des contraintes prédit et reconstruit. Il est également nécessaire d'étudier si l'isotropie de l'apprentissage, c'est-à-dire la capacité du réseau à prédire de façon analogue sur l'ensemble des points du cube, se retrouve dans la prédiction pour des valeurs inédites. Nous allons alors calculer les scores globaux et les scores le long des axes considérant les champs de vitesse filtrés de l'itération 44. Les scores globaux sont renseignés tableau Tab.(4.6); les scores le long des axes sont représentés figures Figs.(4.11). Le tableau montre de très bons résultats sur cette itération inédite. Le tableau fait état de champs prédits à près de 99% corrélés (au sens de Pearson) avec une perte de variance sous les 5% avec également des erreurs relatives en moyenne à 15%. Pour évaluer ces scores il était nécessaire de calculer les équivalents en considérant cette fois les corrélations entre τ_{ij}^{Clark} et τ_{ij}^{FDNS} sur cette même itération; ce sont les valeurs orange du tableau. Ces bons résultats sont confirmés dans les graphes montrant l'erreur relative sur chaque point des trois directions de l'espace. Au passage, on montre que les magnitudes d'erreurs sont sensiblement les mêmes selon les axes confirmant que l'isotropie dans l'apprentissage se retrouve également dans la prédiction de τ_{ij}^{ML} pour des champs de vitesse filtrés inédits. L'écart de précision entre les composantes diagonales et extra-diagonales n'est pas forcément signe que l'IA est plus performante sur les composantes diagonales. En fait, ces dernières composantes font apparaître des gammes de valeurs variées contrairement aux composantes extra-diagonales. Ainsi, pour une précision équivalente, si l'erreur mesurée est de l'ordre de 0.02 (par exemple et dans le sens relatif) elle aura une répercussion plus forte sur ces dernières composantes.

À la fin de ce chapitre, toutes les notions importantes et fondamentales ont été abordées. Le cadre a été posé et le modèle a été entraîné puis évalué. Cette évaluation a été menée pour l'apprentissage puis étendue à des entrées inédites, issues de l'ensemble de test. Il reste à présent à évaluer la capacité de généralisation du réseau. Cette nouvelle évaluation tentera de donner

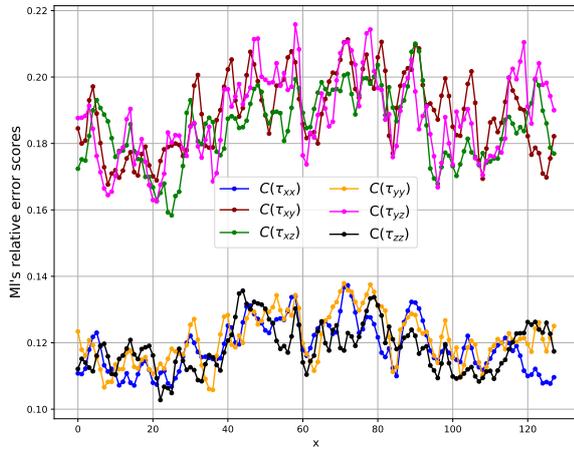
Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.99 / 0.96	0.98 / 0.96	0.98 / 0.96	0.99 / 0.96	0.98 / 0.96	0.99 / 0.96
$1 - \mathcal{R}^2$	0.03 / 0.19	0.03 / 0.09	0.03 / 0.09	0.03 / 0.19	0.03 / 0.09	0.03 / 0.18
E_r	0.12 / 0.31	0.19 / 0.30	0.18 / 0.30	0.12 / 0.31	0.19 / 0.30	0.12 / 0.31

Table 4.6 – À l'itération 44, Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij} .

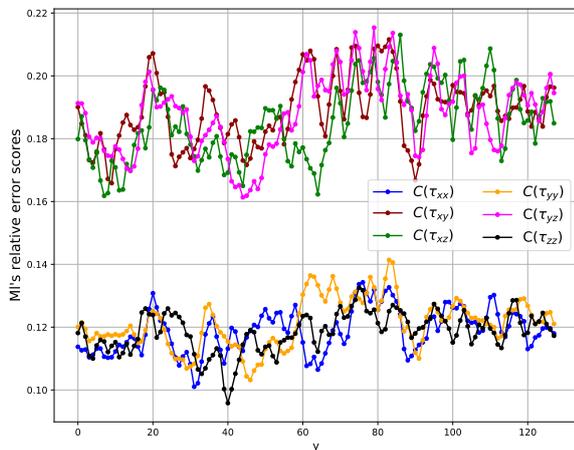
une idée de la gamme d'écoulements sur laquelle le réseau peut être utilisé tout en restant performant. Dans le cadre de l'établissement de modèle pour prédire la Physique non résolue, la notion de performance doit être relativisée avec celle de modèles algébriques existants. De plus, le tenseur en soit n'apparaissant pas dans les équations des grandes échelles, il faut alors évaluer l'impact de la prédiction au niveau vectoriel et au niveau scalaire. En utilisant les corrélations, on peut estimer le degré de justesse entre la divergence de la prédiction et son homologue théorique ; idem pour le niveau scalaire. Ainsi, on aura un ordre d'idée de l'erreur injectée dans les équations de quantités de mouvement ainsi qu'au niveau des transferts énergétiques ou plus précision de la diffusion induite par le terme des petites échelles ainsi que leurs dissipation. Nous donnerons également de manière moins systématique le taux de rétrodiffusion de l'énergie des petites échelles vers les plus grandes qui est souvent une quantité mal appréhendée par les modèles algébriques.

On l'aura compris, ces tests dépassent le cadre de l'intelligence artificielle pure. Le chapitre suivant est donc l'aboutissement de tout ce qui a été présenté ; c'est également la partie principale de notre apport à la communauté scientifique.

(a) Erreurs relatives le long de l'axe x



(b) Erreurs relatives le long de l'axe y



(c) Erreurs relatives le long de l'axe z

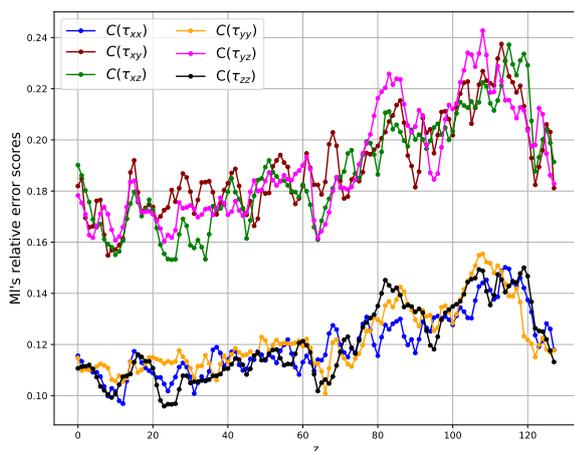
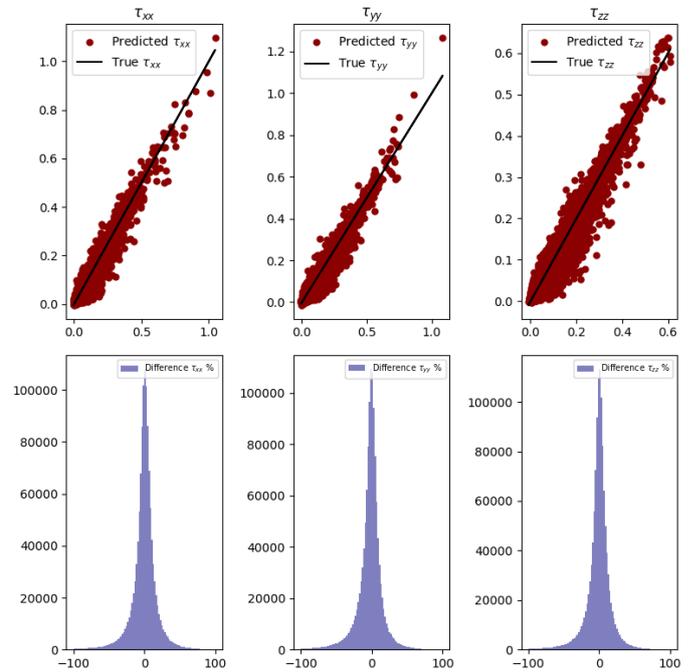


Figure 4.11 – Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales ; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales.

(a) Diagramme de dispersion pour τ_{ii}^{ML}



(b) Diagramme de dispersion pour $\tau_{ij}^{ML} \ i \neq j$

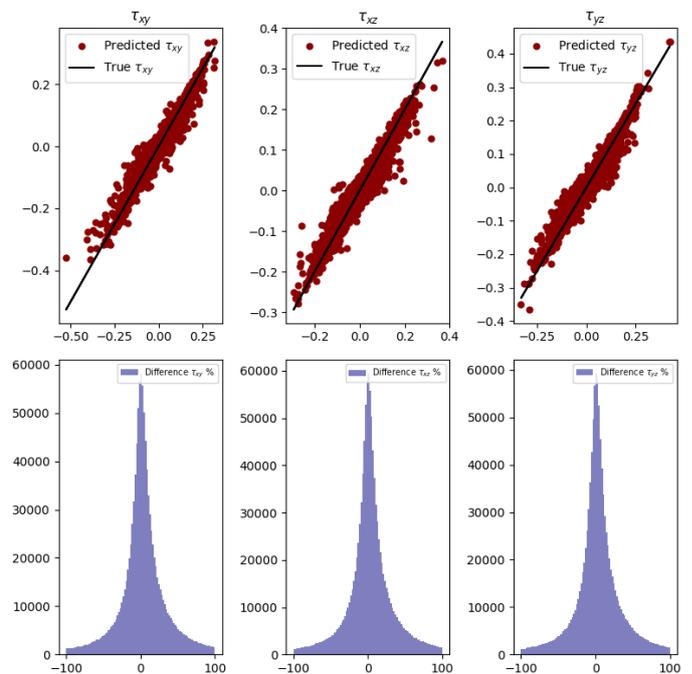


Figure 4.12 – Diagramme de dispersion pour τ_{ij}^{ML}

Chapitre 5

Généralisation de la modélisation du tenseur des contraintes sous mailles par réseaux à convolutions 3D

Au cours du chapitre précédent, nous avons délimité le cadre de notre travail. Nous avons également défini le U-net le plus optimisé au niveau de ses hyper-paramètres mais également la stratégie d'initialisation et d'optimisation des paramètres de ce réseau. Nous avons apporté quelques éléments prouvant que la fonction de coût customisée L_1L_{2M} ainsi que l'optimiseur Nadam étaient les meilleurs choix pour assurer une minimisation des erreurs selon la norme L_1 (écarts en valeur absolue Eq.(3.3.1)) et la norme L_2 (écarts élevés au carré Eq.(3.3.2)), bien que ces choix soient les plus répandus dans la littérature. Ces résultats encouragent à systématiser et varier les explorations de l'espace des hyper-paramètres à la recherche des capacités optimales des architectures et frameworks novateurs existants ou en conception.

Nous avons également testé l'apprentissage du réseau sur des cubes issus de l'ensemble test se conformant au cahier des charges de la validation de l'apprentissage. D'un point de vue Turbulence, nous avons de plus étudié la prédiction et la reconstruction totales à partir des cubes, des 6 composantes du tenseur sous mailles et mesuré la précision des prédictions utilisant les outils introduits dans le chapitre 2 et plus particulièrement dans la section 2.4.2 aux trois niveaux tensoriel, vectoriel et scalaire. Nous avons cherché à répondre à plusieurs questions notamment celle de l'éventualité d'une direction privilégiée au cours de l'apprentissage entraînant potentiellement des écarts importants de qualité de prédiction selon les directions de l'espace. Pour y répondre, nous avons évalué les corrélations de Pearson et les erreurs relatives le long des axes de l'espace. Les conclusions montraient que non seulement aucune direction n'était privilégiée mais également que les prédictions de champs issus de la simulation de référence étaient meilleures que les modélisations des tenseurs τ_{ij} par les modèles algébriques du Gradient et de Smagorinsky.

Dans ce chapitre, nous étendons l'évaluation de notre algorithme sur des simulations plus turbulentes. Nous évaluerons dans un premier temps les scores en volume sur les composantes sous-mailles (niveau tensoriel) et les grandeurs issues de ce tenseur apparaissant dans les équations de Navier-Stokes (niveau vectoriel) et de l'évolution de l'énergie cinétique dans l'écoulement (niveau scalaire). Nous comparerons systématiquement les prédictions avec les

N_{grid}^3	Re_λ	ν	Used for learning
128^3	90	$1e^{-3}$	Yes
176^3	140	$7e^{-4}$	No
256^3	240	$2e^{-4}$	No

Table 5.1 – Caractéristiques des simulations considérées dans ce rapport.

résultats théoriques issus des DNS filtrées¹ (FDNS) ainsi qu’avec les modélisations du modèle de Clark et de façon moins systématique, avec celles issus du modèle Smagorinsky quand cela nous paraîtra nécessaire. De ce fait, nous pouvons évaluer les performances de l’IA de façon absolue et relative à d’autres modèles. Ensuite, nous analyserons les PDF des modélisations algébriques et augmenté par IA en comparaison des données issues de la FDNS. Enfin nous essayerons d’établir l’apport d’information fourni par l’IA sur les interactions L_{ij}^{ML} , C_{ij}^{ML} et R_{ij}^{ML} calculées à partir de τ_{ij}^{ML} conformément aux équations Eqs.(2.4.1), (2.4.2) et (2.4.3), ainsi que leur répercussions énergétiques à l’image de ce qui a été esquissé section 2.4.2. Pour gagner en fluidité et éviter les redondances dans le corps de ce chapitre, le lecteur sera souvent envoyé vers l’annexe B dans laquelle nous compléterons certaines analyses.

Nous présentons les résultats sur deux simulations inédites par rapport à l’apprentissage du réseau, dont le Reynolds sur l’échelle de Taylor en régime développé est respectivement 140 et 240 ; les caractéristiques de ces simulations sont résumées dans le tableau Tab.(5.1). Pour obtenir une description complète des zones de production à celles de dissipation de l’énergie cinétique, en passant par la zone inertielle, il est nécessaire d’augmenter la résolution du domaine. Au passage, de bonnes performances sur des simulations plus résolues seraient synonyme d’une adaptabilité de notre modèle, laissant supposer une compréhension du phénomène latent dans les données, permettant de lier champ de vitesse filtré et interactions sous-mailles.

5.1 Étude des performances du modèle augmenté pour une simulation DNS dont $Re_\lambda \approx 140$

Nous considérons la simulation dont l’évolution temporelle des grandeurs turbulentes est présentée figure Fig.(5.1), le spectre d’énergie en régime développé est présenté figure Fig.(5.2). Nous évaluons la performance de notre modèle à trois stades de l’évolution de la turbulence :

- Au stade du développement des échelles (avant le pic de dissipation) ;
- Aux abords de la zone de stabilisation de l’énergie cinétique et de la dissipation : nous sommes à l’établissement du régime développé ;
- Lorsque la turbulence est complètement développée au sein de l’écoulement.

Chacune des étoiles vertes sur la figure Fig.(5.1) représente l’itération correspondant à chacun de ces stades. À l’avenir ces itérations et les stades de la turbulence associés seront désignées par leur numéro respectif 42, 88 et 118 (en centaine).

1. Rappelons que par défaut la DNS est filtrée tel que $R = \kappa_{\text{max}}/\kappa_C = 2$

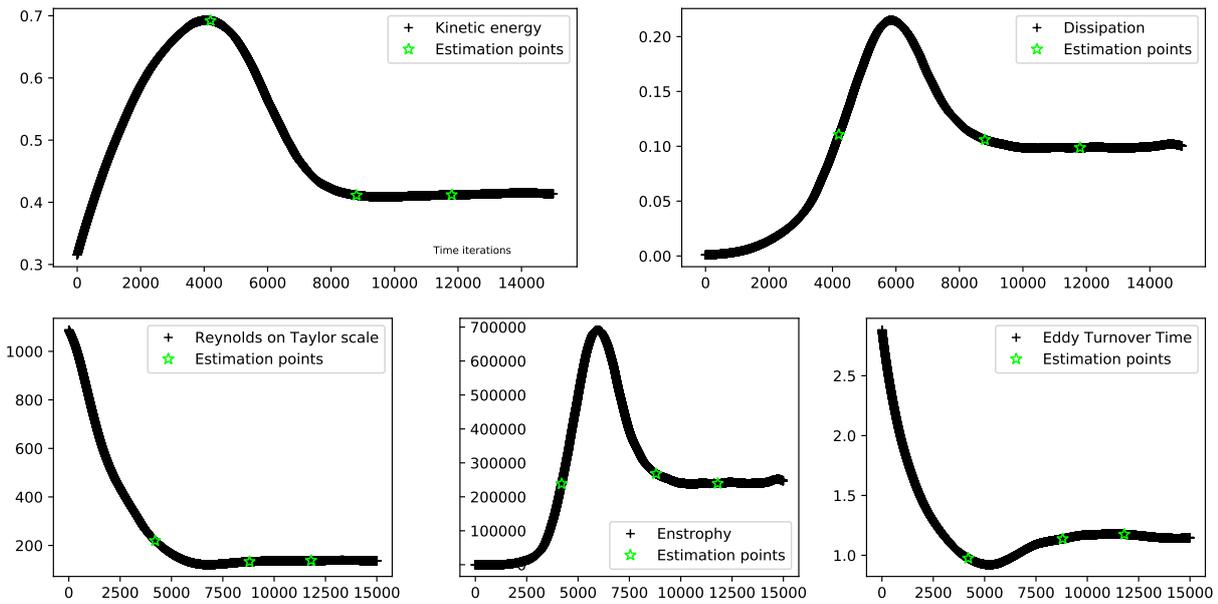


Figure 5.1 – Évolution temporelle des grandeurs caractéristiques de la Turbulence pour la simulation dont la résolution est $N = 176$ et $Re_\lambda \approx 140$ dans le régime développé.

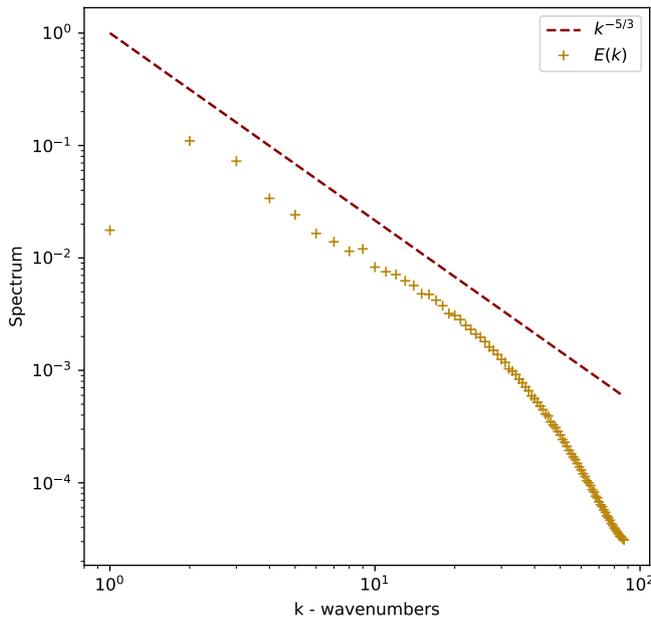


Figure 5.2 – Ci-contre - Spectre de l'énergie cinétique calculée dans le régime développé de la turbulence au sein de l'écoulement issue de la simulation au Reynolds $Re_\lambda \approx 140$.

Attentes des performances de l'IA : opposition de point de vue - Avant de tester les performances sur ces itérations, il est difficile de déterminer le régime pour lequel espérer les meilleures performances. De prime abord, puisque l'IA a été entraînée sur des champs de turbulence développée, l'on s'attend à ce que les performances du réseau soient les meilleures pour l'itération 118. D'un autre côté, c'est dans le régime développé que les échelles sont les plus variées et puisque le Reynolds est plus grand et que la résolution est plus fine pour cette simulation, on conjecturerait, qu'au contraire, les champs de vitesse les plus proches de ceux

utilisés pour l'apprentissage étant ceux des itérations avant l'établissement de la turbulence, les performances les plus optimales seront obtenues pour les itérations *avant* le régime développé de cette nouvelle simulation. Nous illustrons les différences entre champs de vitesse considérés lors de l'entraînement et ceux considérés dans cette section, figure Fig.(5.3). Ces deux façons d'analyser cette problématique correspondent à deux approches radicalement différentes : celle de l'IA et celle de la Physique. Les régimes turbulents sont définis par des critères qui sont étrangers à l'intelligence artificielle telle que nous l'avons construite. D'un point de vue IA, l'important est l'information contenue dans l'entrée peu importe sa nature Physique ou son positionnement dans le stade de l'évolution de la turbulence. Ainsi pour assurer la cohérence entre l'entraînement et la prédiction nous nous sommes assurés que ces points de vue étaient respectés d'une part en filtrant les écoulements dans la zone inertielle du spectre d'énergie cinétique, et

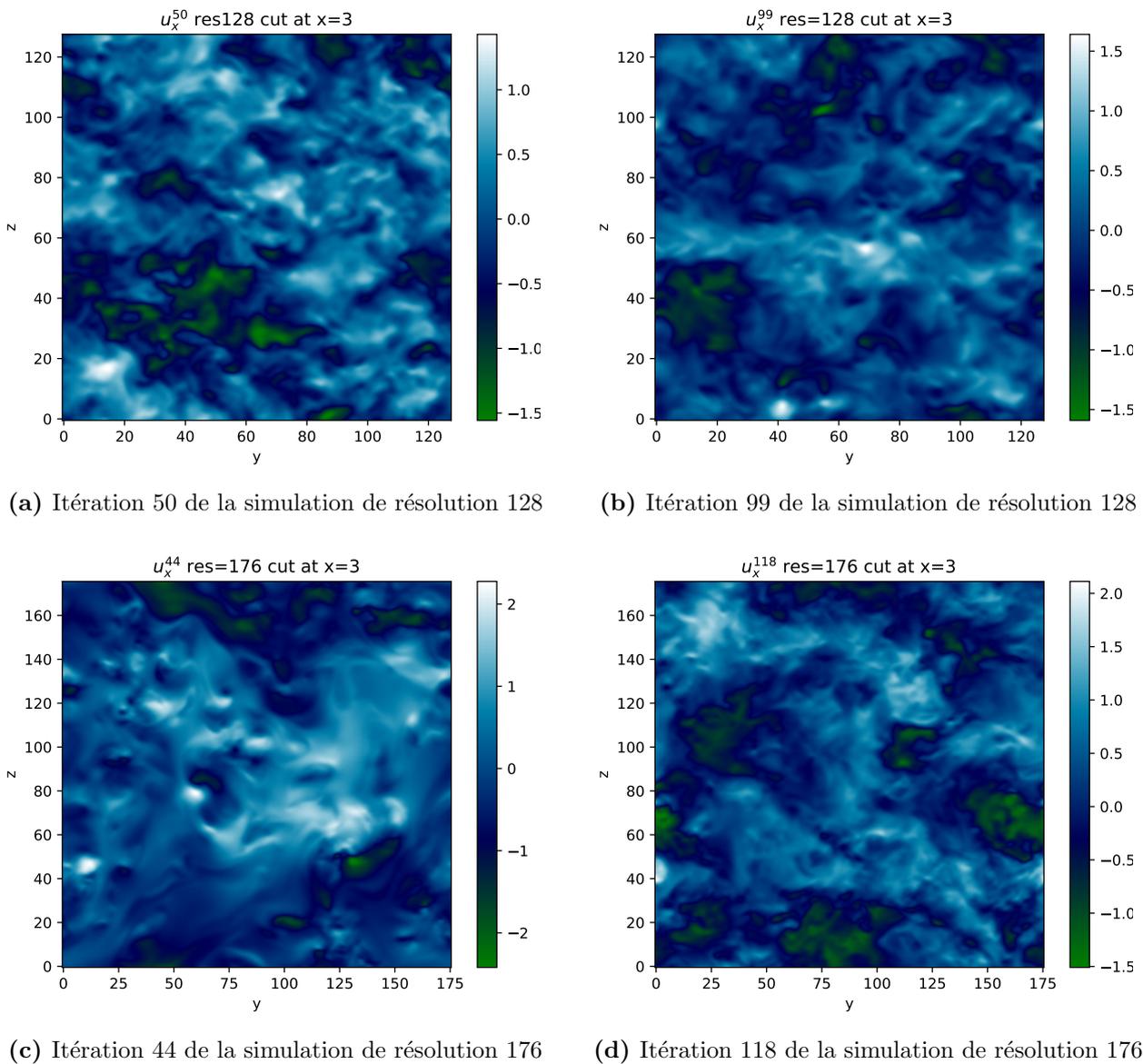


Figure 5.3 – Illustrations des premiers et derniers u_x balayés lors de l'apprentissage (figures du haut) et lors des tests de cette section de (figures du bas) Voir tableua Tab.(5.1).

d'autre part en s'assurant que les cubes de champ de vitesse renseignent assez d'information sur la dynamique et contiennent environ 90% des transferts énergétiques à travers la coupure (voir section 4.2.1).

Dans la suite, nous adoptons le point de vue de la turbulence, c'est-à-dire que nous considérons que puisque le régime de turbulence développée met en jeu des processus universels, nous supposons que l'IA fournira des prédictions précises des composantes du tenseur des contraintes sous mailles pour le régime turbulent, dont elle a déjà une expérience. Nous cherchons à évaluer à travers l'itération 118 si les changements de résolution, de viscosité et de Reynolds (voir Tab.(5.1)) entravent la généralisation sur une turbulence développée et établie. Ce résultat est un préliminaire à l'étude de la généralisation et permet de distinguer montée en Reynolds et régime d'écoulement différent.

5.1.1 Impact de la résolution sur les prédictions en régime développé

Nous débutons notre analyse par le calcul des scores (Pearson, \mathcal{R}^2 et l'erreur relative) entre prédictions et champs issus de la FDNS pour l'itération 118 la plus proche de l'ensemble d'entraînement. Les résultats sont rapportés dans le tableau Tab.(5.2). Les couleurs des valeurs du tableau sont utilisées pour distinguer le modèle sur lequel sont calculés ces scores : en noir nous renseignons les performances du modèle augmenté ; alors qu'en orange le modèle de Clark. Les valeurs renseignées dans ce tableau concernent la prédiction des composantes τ_{ij} (niveau tensoriel). Notons enfin, que les scores sont calculés sur l'ensemble du champ reconstitués à partir du découpage en cube du champ de vitesse.

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.99 / 0.97	0.99 / 0.97	0.99 / 0.97	0.99 / 0.96	0.99 / 0.97	0.99 / 0.97
$1 - \mathcal{R}^2$	0.02 / 0.12	0.03 / 0.07	0.03 / 0.07	0.02 / 0.13	0.03 / 0.07	0.02 / 0.13
E_r	0.11 / 0.28	0.17 / 0.27	0.17 / 0.26	0.11 / 0.28	0.17 / 0.27	0.11 / 0.28

Table 5.2 – Scores obtenus sur la prédiction au niveau tensoriel à l'itération 118. Les valeurs en noir et en orange représentent resp. les scores obtenus par l'IA et par le modèle Clark sur τ_{ij} .

Au niveau tensoriel, les résultats (Tab.(5.2)) sont très satisfaisants. Non seulement les performances du réseau ne sont impactées ni par le changement de résolution intrinsèque à la simulation, ni par le changement du Reynolds, mais elles sont également meilleures que celles issues de la modélisation de Clark dont les corrélations de Pearson sont notablement hautes. On remarque également la moindre perte de la variance (moment d'ordre 2) et en particulier

2. Le score \mathcal{R}^2 sera utilisé pour évaluer la variance manquante entre prédiction/modélisation et FDNS. Nous calculerons donc $1 - \mathcal{R}^2$ comme précisé section 2.4.2

pour les composantes diagonales : le modèle du Gradient perd 13% de la variance là où les composantes obtenues par IA n'en perdent que 2%. Enfin, l'erreur relative met en exergue le fait que le ML prédit avec plus de précision (relative) les composantes diagonales, nous l'avons déjà remarqué dans l'évaluation de l'apprentissage section 4.4. Rappelons que cela n'implique pas que dans l'absolu l'IA performe mieux sur la prédiction des composantes diagonales, mais que relativement aux champs et leurs magnitudes, le pourcentage d'erreur est moins élevé pour les composantes diagonales. Nous continuons notre analyse en volume aux niveaux vectoriel et scalaire concentrant de plus en plus l'information prenant en compte les variations dans l'espace du tenseur τ^{ML} et le flux résultant (à travers l'opérateur divergence), ou bien ses interactions avec d'autres grandeurs turbulentes comme le champ de vitesse ou de déformations respectivement \overline{U}_i et \overline{S}_{ij} tous deux issus de la FDNS pour ces tests a priori. Les scores obtenus sont rapportés dans le tableau Tab.(5.3). Les trois premières colonnes du tableau représentent les composantes de la divergence du tenseur sous mailles qui traduit la diffusion de la quantité de mouvement portée par les composantes du tenseur. À ce niveau vectoriel, les champs calculés à partir de τ^{DNS} recouvrent 90% de la variance présente dans les données FDNS et sont corrélés à 95% avec ces dernières. Les différences avec les quantités estimées à partir de la modélisation Clark sont significatives. Les deux autres colonnes représentent les deux scalaires Π et $\Psi = \mathbf{U} \cdot (\nabla \cdot \tau)$ (comme précisé section 2.4.1) permettant d'estimer l'impact énergétique des erreurs des prédictions de τ^{ML} , que ce soit au niveau de la dissipation de l'énergie cinétique par les échelles non résolues (Π) ou la diffusion de cette énergie par interactions entre échelles résolues et non résolues (Ψ).

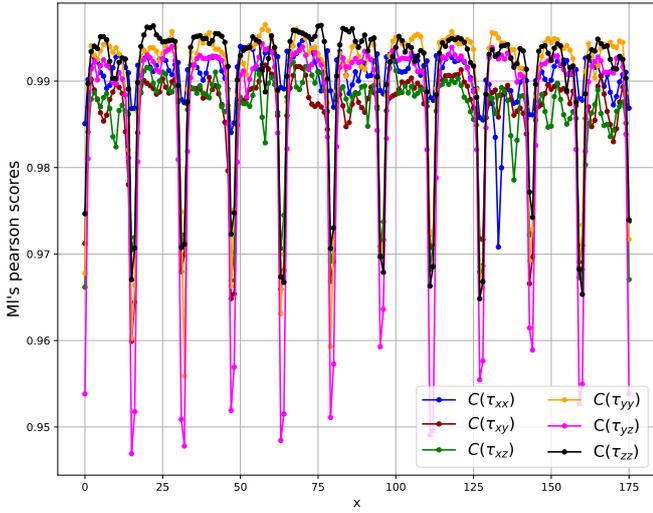
Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.94 / 0.89	0.95 / 0.89	0.95 / 0.87	0.98 / 0.95	0.95 / 0.89
$1 - R^2$	0.11 / 0.21	0.10 / 0.22	0.11 / 0.21	0.04 / 0.10	0.10 / 0.21
E_r	0.33 / 0.47	0.33 / 0.46	0.33 / 0.46	0.19 / 0.30	0.31 / 0.45

Table 5.3 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 118.

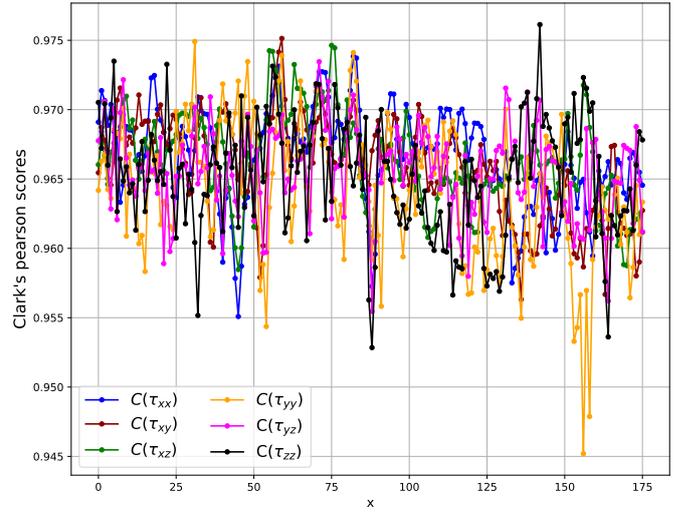
Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes $\partial_{x_j}\tau_{ij}$ (niveau vectoriel) et les grandeurs du niveau scalaire Π et Ψ

A l'issue de ces évaluations en volume, nous pouvons argumenter que le réseau est globalement insensible au changement de résolution intrinsèque aux simulations d'où proviennent les entrées qu'il traite. Nous retrouvons la supériorité de notre modèle par rapport au modèle de Clark que ce soit au niveau de la prédiction des composantes du tenseur τ mais également des conséquences de cette dernière sur l'évolution du champ de vitesse et sur le bilan énergétique. La prédiction permet de recouvrer également près de 98%, 90% et 93% de la variance contenue en moyenne dans les niveaux tensoriel, vectoriel et scalaire des grandeurs FDNS concernées, performances notablement plus appréciables que celles du modèles Clark.

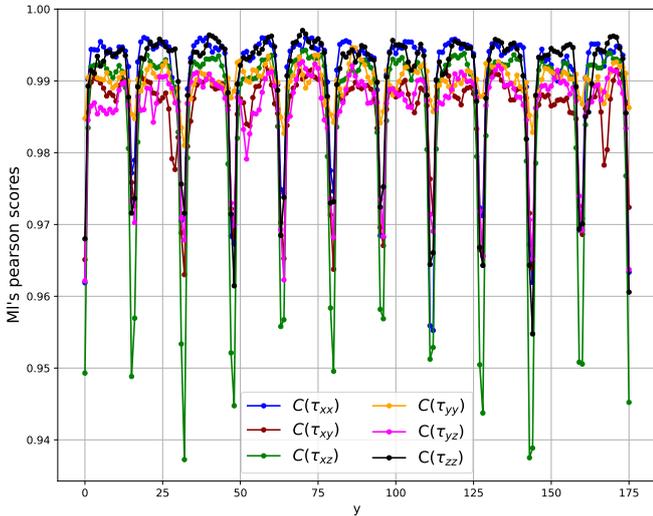
(a) Score de Pearson le long de l'axe x (ML)



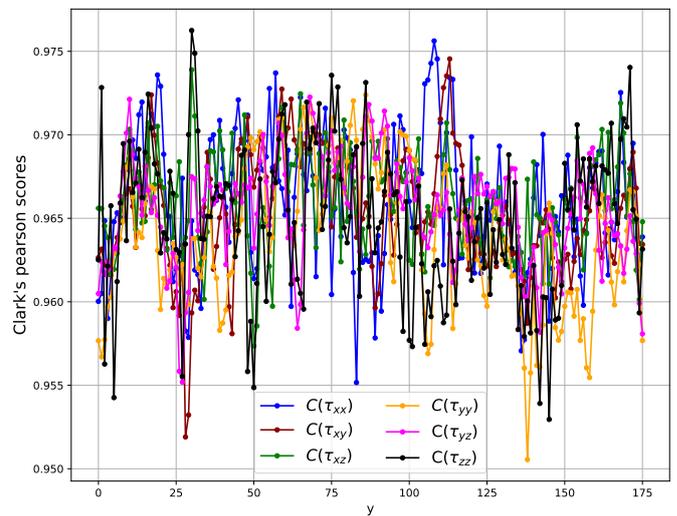
(b) Score de Pearson le long de l'axe x (Clark)



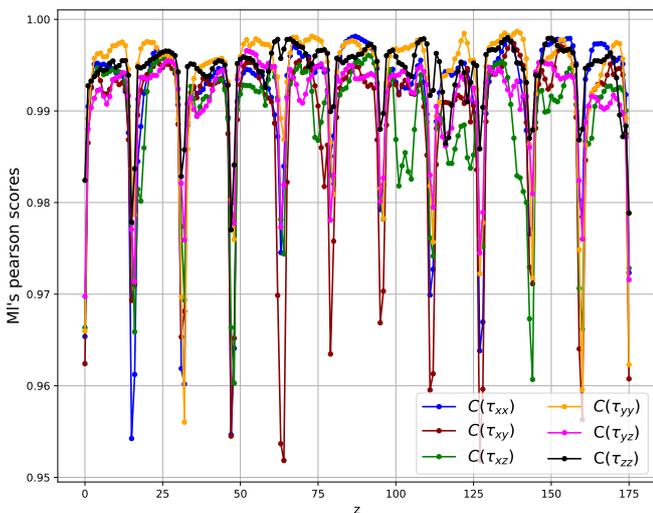
(c) Score de Pearson le long de l'axe y (ML)



(d) Score de Pearson le long de l'axe y (Clark)



(e) Score de Pearson le long de l'axe z (ML)



(f) Score de Pearson le long de l'axe z (Clark)

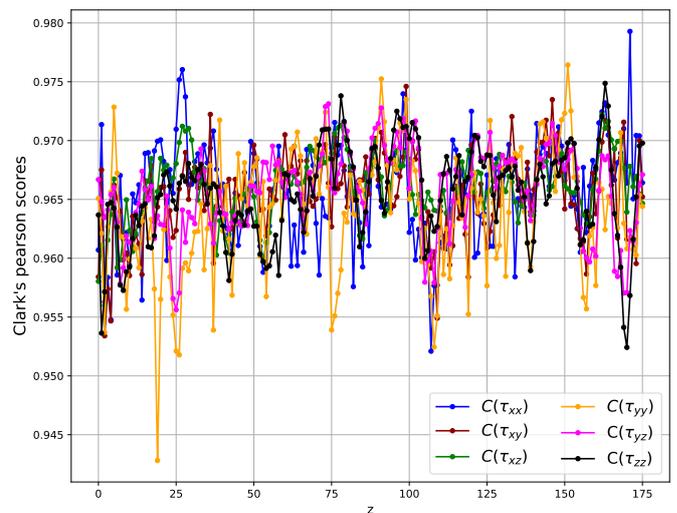


Figure 5.4 – Corrélations de Pearson entre les composantes de τ_{ij}^{FDNS} et celles de τ_{ij}^{ML} (figures à gauche). Idem avec les composantes de τ_{ij}^{Clark} (figures à droite) à la place de la prédiction.

Nous testons à présent l'isotropie de la généralisation en calculant le score de Pearson sur τ_{ij} le long des directions de l'espace (en complément, les erreurs relatives sont rapportées dans figures Fig.(B.2)). Nous calculons et reportons Figures Figs.(5.4a), (5.4c) et (5.4e) les corrélations de Pearson entre τ_{ij}^{ML} et τ_{ij}^{DNS} le long des axes x , y et z ; de même, nous traçons les corrélations de Pearson entre τ_{ij}^{Clark} et τ_{ij}^{DNS} figures Figs.(5.4b), (5.4d) et (5.4f). À la lumière de ces figures nous pouvons établir plusieurs observations. Tout d'abord nous remarquons la présence des pics de baisse de précision locale dont nous avons discuté brièvement section 4.4. Cette baisse de précision s'étend sur les points aux bords de chaque cube et est la conséquence de la façon dont les convolutions glissent le long des entrées dans un CNN. Fatalement, les points aux bords sont moins considérés que les points successifs. Notons que la stratégie de padding mise en place permet de limiter la baisse de précision sur les points limitant les cubes, c'est-à-dire les six surfaces de N_c^2 points. On peut estimer N_{impact} le nombre de points sur lesquels la précision du CNN seront moins précises : avec N le nombre de points de discrétisation de la simulation, :

$$N_{\text{impact}} = 6N_c^2 (N/N_c)^3 \quad (5.1.1)$$

Ramené sur l'ensemble des points du volume, le nombre de points impactés représente approximativement 40 % des points de la simulation. En dehors de ces points, la prédiction est globalement meilleure que la modélisation de Clark et ce pour toutes les composantes τ_{ij} avec des corrélations au dessus de 99%. Au niveau global, ces pertes de précision ne se font presque pas ressentir comme on le verra figure Fig.(5.6). Ceci nous amène à la deuxième constatation qui concerne notamment l'isotropie de l'apprentissage. On remarque que les composantes diagonales (τ_{xx} , τ_{yy} et τ_{zz} respectivement représentées par les courbes bleue, orange et noire) sont globalement (et de façon relative) mieux apprises que les composantes extra-diagonales (τ_{xy} , τ_{xz} et τ_{yz} respectivement représentées par les courbes bordeaux, verte et rose). À propos de ces dernières, les pics sont plus prononcés. Il semble également que selon l'axe de projection, la composante extra-diagonale des contraintes sous mailles perpendiculaire à cette direction présente des pics plus prononcés. Si on regarde la figure Fig.(5.4a) par exemple, la composante τ_{yz} sera la moins précise au niveau des pics. Mais cette remarque n'est en fait valable que pour les points aux bords des cubes, dans cette même figure on voit que loin de ces bords (c'est-à-dire au point adjacent), les performances sont identiques que sur les autres composantes extra-diagonales. Nous pouvons donc confirmer que le caractère isotrope de l'apprentissage mis en exergue section 4.4 a été généralisé par l'IA. En d'autres termes, il n'existe pas de direction d'apprentissage privilégiée au sein du cube. Ainsi par la suite, il ne sera pas nécessaire de calculer les scores le long des trois axes, un seul axe suffira.

Nous continuons l'analyse des capacités de généralisation sur ce régime développé en traçant les PDFs des τ_{ij} pour la FDNS (courbe noire continue), la prédiction par Machine Learning (ligne brisée rouge), et les modélisation τ_{ij}^{Clark} et τ_{ij}^{Smago} (respectivement courbes orange et vertes en pointillé) dans les figures Fig.(5.5). Chacune de ces PDFs sont standardisées par les moyenne et déviation de la FDNS. L'axe des ordonnées est en échelle logarithmique pour prendre en compte une gamme de fréquences d'événements plus étendue. Comme nous l'avons mentionné, les PDFs expriment de façon continue la densité de chaque événement, ou valeur, dans le champ total. Par la densité d'événement, on entend la fréquence d'un événement ramenée sur le nombre d'événements totaux, divisée par le pas de discrétisation entre chaque événement. Dans notre cas, le nombre d'intervalles de l'histogramme est $b = 400$; la gamme de valeurs dépendra de la

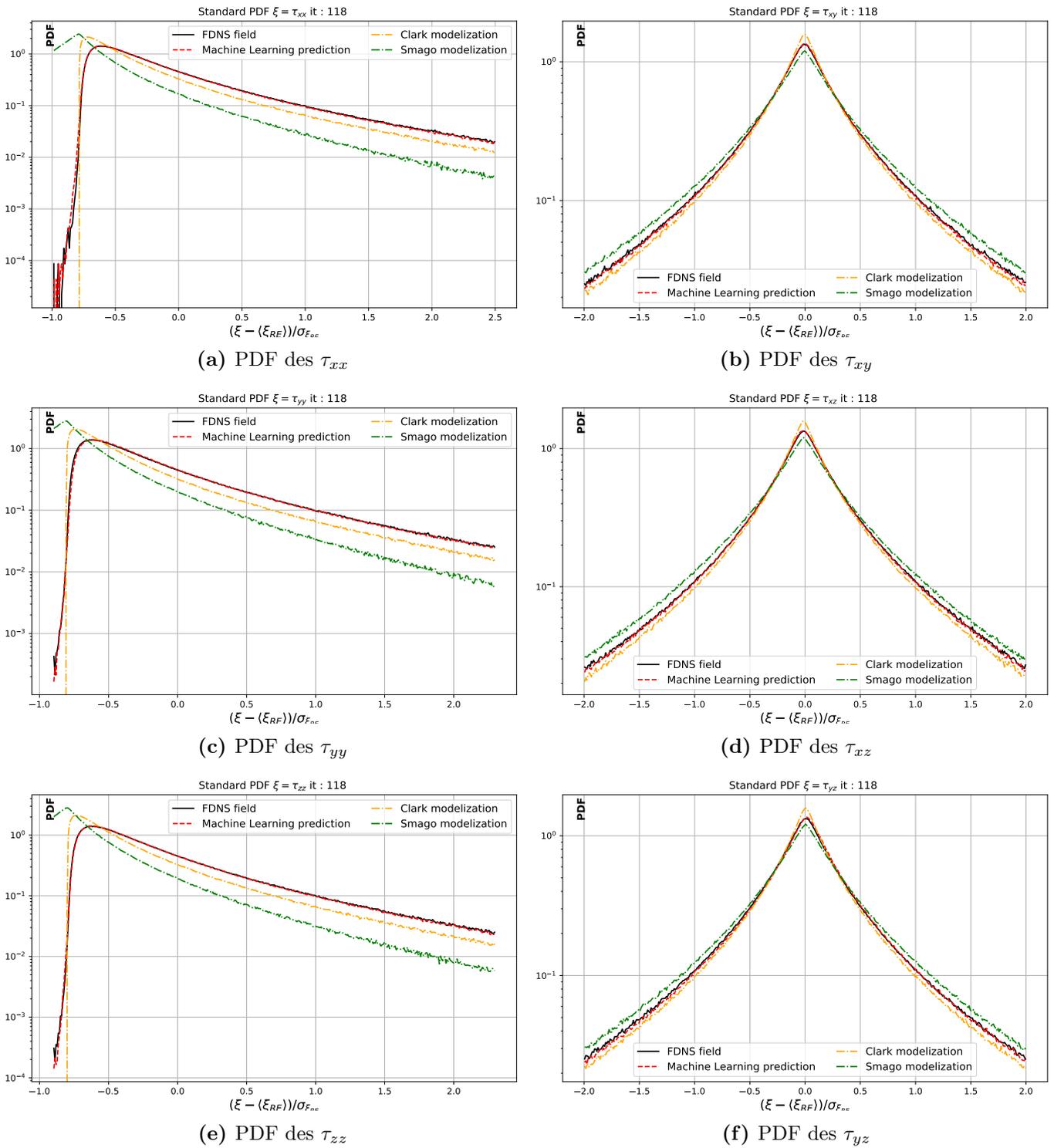


Figure 5.5 – À l’itération 118, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

grandeur voire de la composante.

On peut établir deux constats à partir des figures Figs.(5.5). Dans un premier temps, on constate

que la justesse des distributions issues des modèles Clark et Smagorinsky oscillent entre très similaires et très différentes, en particulier pour le deuxième modèle selon si les composantes sont extra-diagonales ou diagonales (resp.). Au contraire, l'IA prédit des signaux dont les distributions sont fidèles aux champs cibles de la FDNS de façon homogène. Au niveau des composantes diagonales et extra-diagonales, les événements les plus rares sont également mieux représentés par la prédiction que les autres modèles. Ces événements sont généralement intenses et traduisent l'intermittence. Le fait que l'on retrouve cette intermittence (située donc aux queues des distributions) renseigne sur la bonne minimisation de l'écart au sens de la norme L_1 , entre FDNS et prédictions par IA. La juxtaposition au niveau des événements plus courants renseigne sur l'écart au carré (L_2). Notons que les PDFs ne tiennent pas compte de la répartition des différences dans l'espace. Les meilleurs exemples sont les PDFs issues du modèle de Smagorinsky. En effet, alors que les distributions des composantes extra-diagonales issues de ce modèle sont très similaires à celles de la FDNS, les corrélations de Pearson sont autour de 24%. On peut constater ces erreurs dans les champs en eux mêmes. Nous traçons les coupes des grandeurs τ_{xy}^{FDNS} et τ_{xy}^{Smago} dans le plan $x = 3$ figures Figs.(5.6). Dans ces figures, et en particulier les figures du haut, les échelles ne sont pas fixées. On voit que les gammes de magnitudes pour la modélisation de Smagorinsky (figure du haut à droite) sont les mêmes que celles de la FDNS (figure du haut à gauche). Ainsi, il faudra toujours combiner les PDFs avec les corrélations de Pearson (entre autres) et des visualisations directes quand cela est nécessaire.

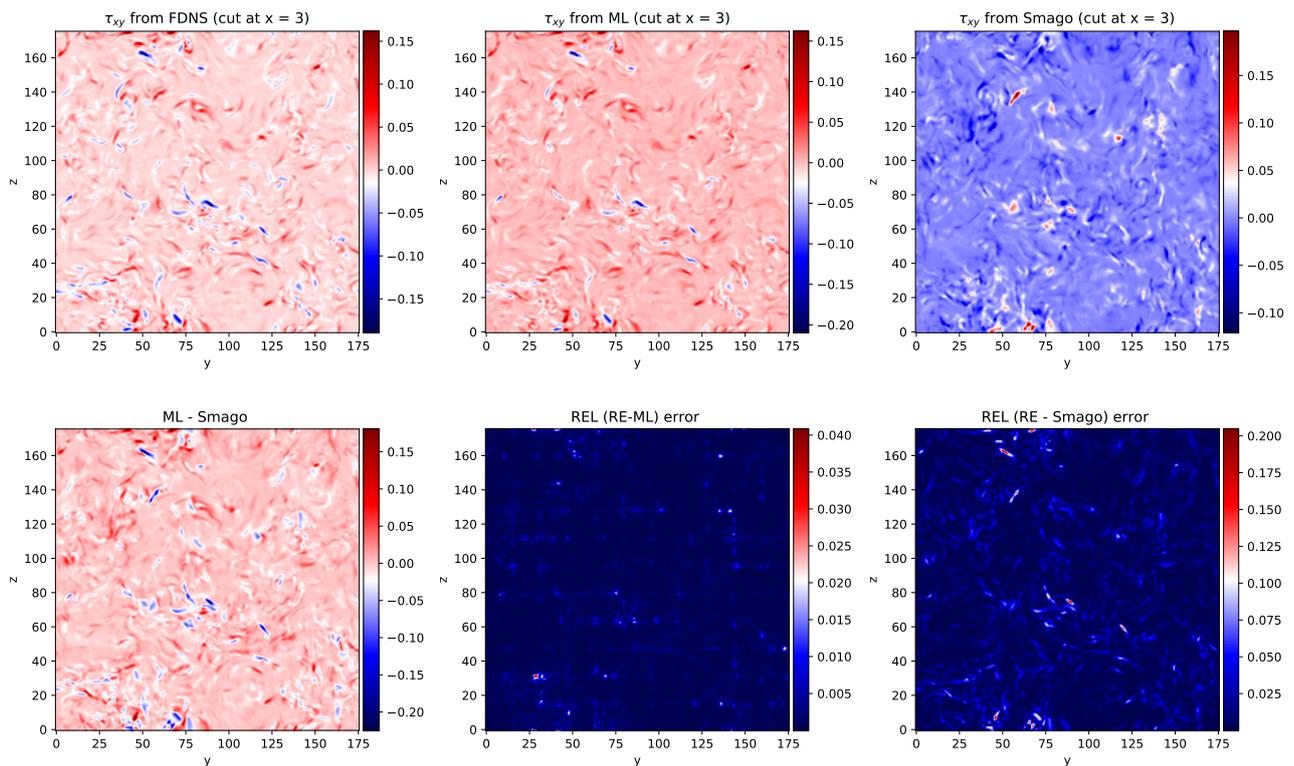


Figure 5.6 – **Figures du haut** : Coupe de la composante τ_{xy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Smagorinsky (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Smagorinsky et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$. Les échelles sont volontairement propres à chacune des coupes.

Nous voulons à présent étudier les distributions des grandeurs prédites et modélisées aux niveaux vectoriel et scalaire. Nous traçons alors les PDFs des trois composantes du vecteur $\nabla \cdot \boldsymbol{\tau}$ ainsi que celle du scalaire Π figures Figs.(5.7). À ces niveaux, on constate tout d'abord que le Smagorinsky ne prédit plus les bonnes distributions comme au niveau tensoriel, alors que paradoxalement les corrélations de Pearson entre $(\nabla \cdot \boldsymbol{\tau})^{\text{Smago}}$ et $(\nabla \cdot \boldsymbol{\tau})^{\text{FDNS}}$ sont proches de 50 % en moyenne. On peut supposer que les erreurs des composantes tensorielles ont contaminé les résultats au travers des contractions tensorielles. Nous établissons le même constat au niveau scalaire où une bonne partie des événements n'apparaissent pas dans la distribution.

Dans un second temps, les distributions des grandeurs construites à partir de $\boldsymbol{\tau}^{\text{ML}}$ et $\boldsymbol{\tau}^{\text{Clark}}$ sont assez proches de celles basées sur $\boldsymbol{\tau}^{\text{FDNS}}$. L'IA semble même plus précise et couvre mieux les événements courants comme les plus rares. Les PDFs comparatives sur Ψ sont tracées dans la section B.1.1 (figure Fig.(B.1)) de l'annexe associé à ce chapitre.

Nous traçons les coupes des composantes z des vecteurs $(\nabla \cdot \boldsymbol{\tau})^{\text{FDNS}}$ et $(\nabla \cdot \boldsymbol{\tau})^{\text{Clark}}$ dans le plan $x = 3$. Cette fois ci, les échelles de couleurs sont fixées à celles de la FDNS. Les coupes en elles mêmes sont représentées dans les figures du haut ; on peut voir que la prédiction (au centre) et

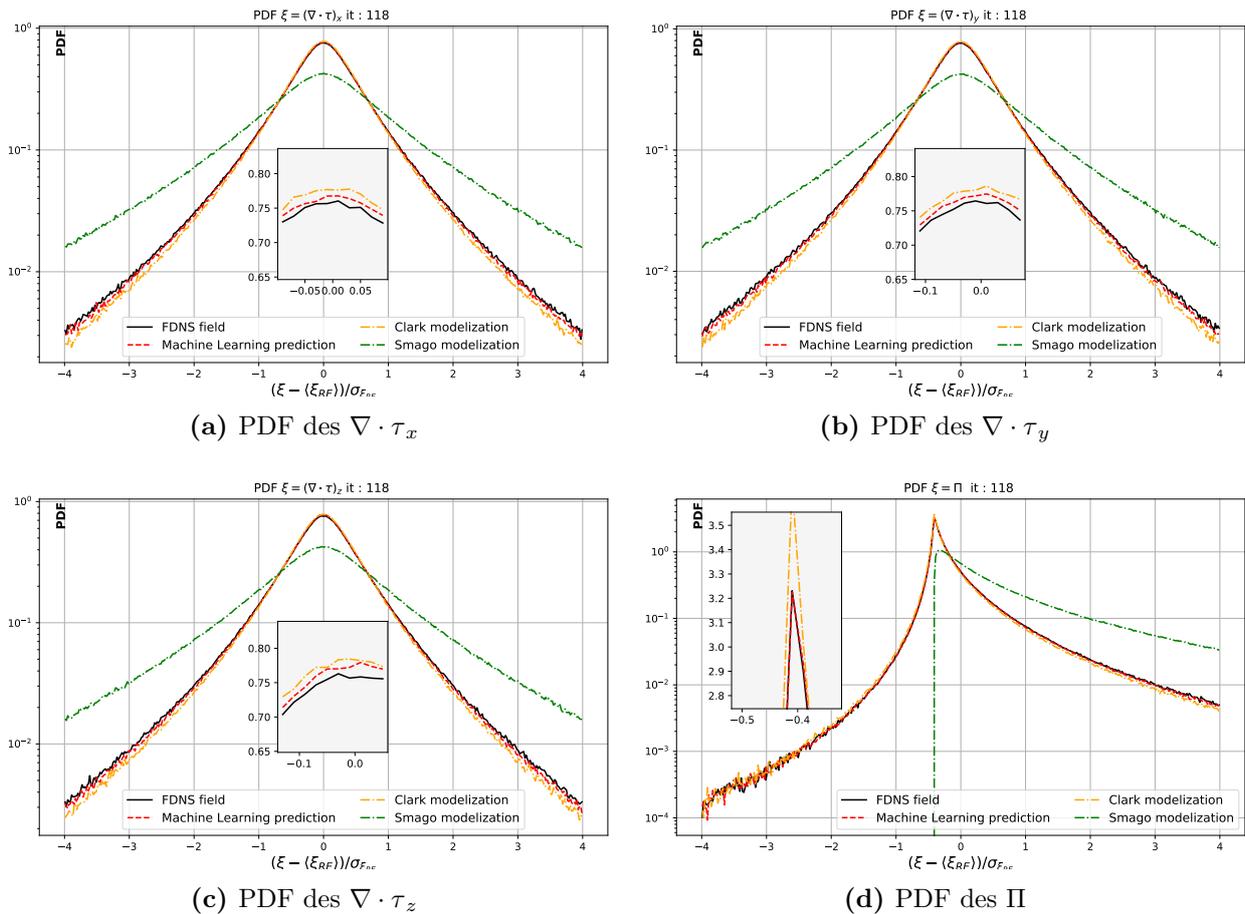


Figure 5.7 – À l'itération 118, comparaisons des PDFs des composantes du vecteur $\partial_{x_j} \tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenus à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte) ; Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

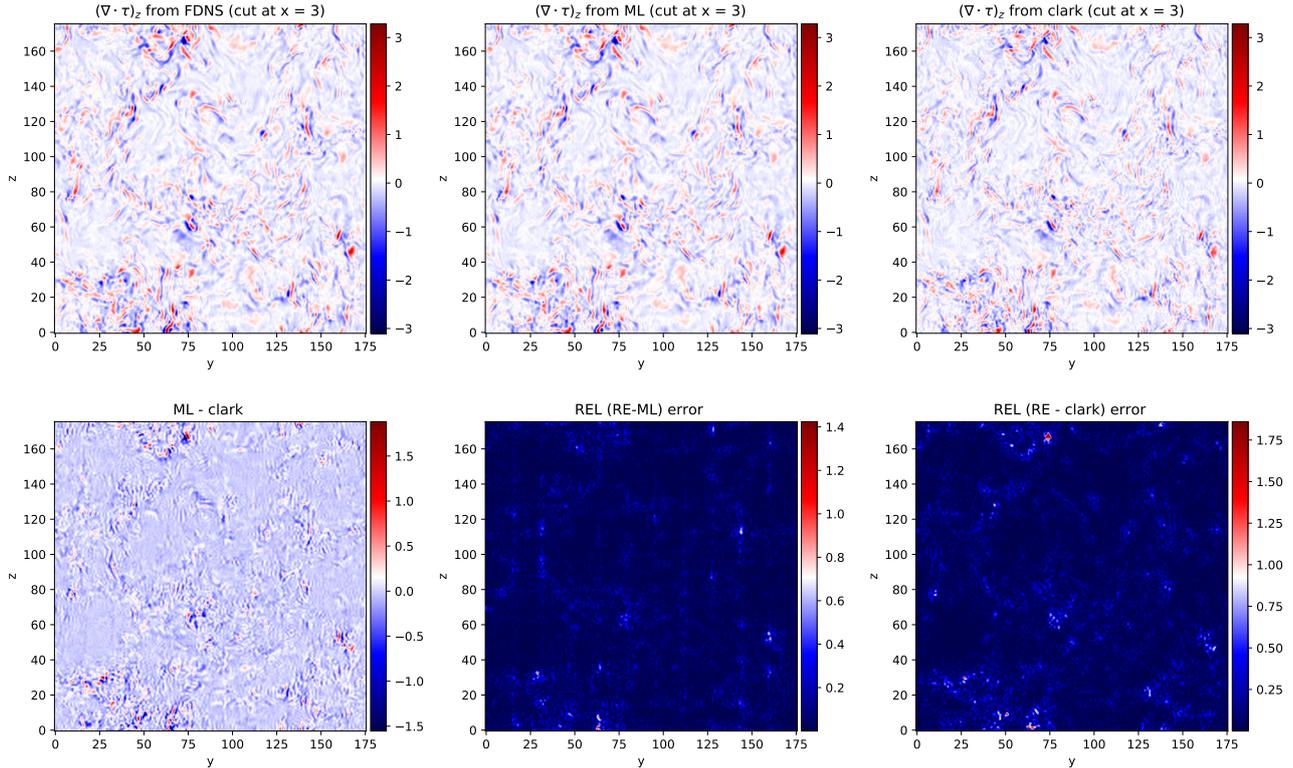


Figure 5.8 – **Figures du haut** : Coupe de la composante $(\nabla \cdot \tau)_z$ FDNS (à gauche), prédite par IA (au centre), et obtenue par le modèle de Clark (à droite). Les échelles de couleurs sont celles de la FDNS. **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$, à l’itération 118.

la modélisation par le modèle de Clark (à droite) sont assez semblables. Les écarts en valeurs absolues, figures du bas, entre la prédiction et la FDNS (au centre) sont globalement moins diffus que ceux existant dans le tracé des erreurs entre la modélisation et FDNS (à droite). Ces erreurs sont également de plus faibles amplitudes et on remarque qu’elles sont sensiblement localisées au bord des cubes sur lesquels les prédictions sont faites. Ces figures soulignent enfin que l’indépendance des cubes et de leur précision intrinsèque n’impactent pas les champs τ_{ij} reconstruits y compris au niveau vectoriel et scalaire.

Dans cette section, il a été établi que la généralisation est isotrope à l’instar de l’apprentissage. Le changement de résolution ainsi que la montée en Reynolds de près 60% par rapport à la simulation utilisée pour générer l’ensemble d’apprentissage n’ont pas affecté les performances de l’IA surpassant sur tous les points le modèle de Clark. Nous avons également pointé du doigt le fait que les prédictions sur les points exactement au bord de chaque cube étaient moins bonnes qu’aux autres positions. L’impact ne se fait pas ressentir au niveau de la performance globale ni dans la construction de grandeurs dépendantes du tenseur τ_{ij}^{ML} comme sa divergence ou sa contraction avec le tenseur des déformations filtré \overline{S}_{ij} pour la constructions de la dissipation de l’énergie cinétique par les échelles non résolues.

Nous voulons à présent tester la capacité d’adaptation de l’IA lorsque le champ de vitesse

présente des structures qu'il a peu vu durant l'apprentissage, voire pas du tout. Le premier cas est représenté par l'itération 88 qui est l'objet de la section suivante. Le deuxième cas sera l'objet du paragraphe 5.1.3.

5.1.2 Prédiction à l'entrée du régime développé : étude des composantes de la décomposition de Leonard

Comme nous allons le montrer, les performances du réseau de neurones à ce stade de développement de la Turbulence (itération 88) sont très similaires à ceux de l'itération précédente pour laquelle la turbulence était pleinement développée. Comme précédemment, nous rapportons les scores globaux dans le tableau Tab.(5.4)

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.99 / 0.96					
$1 - \mathcal{R}^2$	0.02 / 0.13	0.03 / 0.07	0.02 / 0.07	0.02 / 0.14	0.03 / 0.07	0.02 / 0.14
E_r	0.11 / 0.28	0.17 / 0.27	0.17 / 0.27	0.11 / 0.28	0.17 / 0.27	0.11 / 0.28

Table 5.4 – Scores obtenus sur la prédiction au niveau tensoriel à l'itération 88. Les valeurs en noir et en orange représentent resp. les scores obtenus par l'IA et par le modèle Clark sur τ_{ij} .

Nous pouvons alors valider les conclusions faites pour l'itération précédente. Les prédictions sont globalement plus précises que le modèle de Clark et plus d'information est recouvrée. Ces résultats permettent de supposer que les distributions des composantes τ_{ij}^{ML} seront très proches de celles de τ_{ij}^{FDNS} . Nous reportons Tab.(5.5) les scores obtenus pour les grandeurs correspondant aux niveaux vectoriel et scalaire construites à partir de τ_{ij}^{ML} et de τ_{ij}^{Clark} par rapport à celles construites à partir de τ_{ij}^{FDNS} . Dans ce tableau aussi, les scores sont très proches de ceux obtenus pour l'itération précédente. Cela confirme que les erreurs de la prédiction majoritairement concentrées aux bords des cubes impactent moins les grandeurs vectorielles et scalaires que les erreurs commises par le modèle de Clark sur τ_{ij} . En d'autres termes, un modèle sous-maille dont la modélisation de τ_{ij} impliquerait l'IA présentée ici fournirait une modélisation des échelles non résolues supérieures que celles fournies par le modèle de Clark au niveau de la prédiction en soi, ainsi qu'au niveau de ses conséquences sur l'évolution des échelles résolues, et de la quantité de mouvement que de l'énergie cinétique. Cette conclusion est pour l'instant valable pour le régime développé de la turbulence pour une gamme de Reynolds Re_λ compris entre 90 et 140. Globalement, les PDFs des composantes du tenseur sous-mailles, de sa divergence, et de la dissipation énergétique par les échelles non résolues sont très proches voire similaires au cas précédent. Nous les traçons toutes dans l'annexe section B.1.2.

Nous proposons une étude sur les grandeurs L_{ij}^{mod} , C_{ij}^{mod} et R_{ij}^{mod} définies par les équations (2.4.1),

Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.94 / 0.88	0.94 / 0.88	0.94 / 0.88	0.98 / 0.95	0.95 / 0.89
$1 - R^2$	0.15 / 0.22	0.14 / 0.22	0.14 / 0.22	0.05 / 0.11	0.10 / 0.22
E_r	0.33 / 0.47	0.33 / 0.47	0.33 / 0.47	0.20 / 0.30	0.32 / 0.46

Table 5.5 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l’itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l’IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j} \tau_{ij}$ et des grandeurs scalaires Π et Ψ

(2.4.2) et (2.4.3). L’idée est d’essayer de comprendre physiquement la meilleure performance de l’IA par rapport au modèle de Clark. En fait, on essaye de déterminer les interactions sous mailles mieux modélisées par l’IA. Rappelons que cette analyse n’est pas exacte puisque la façon dont nous construisons les grandeurs L_{ij} , C_{ij} et R_{ij} n’assure pas forcément l’identité

$$\tau_{ij}^{\text{Mod}} \neq L_{ij}^{\text{Mod}} + C_{ij}^{\text{Mod}} + R_{ij}^{\text{Mod}}$$

mais comme nous l’avons déjà mentionné section 2.4.1, cette étude permet d’avoir une idée de l’erreur commise par la modélisation sur chacune des composantes **indépendamment** les unes des autres. Avant de plonger dans cette étude, mentionnons que les distributions des composantes diagonales sont très similaires entre elles, idem pour les composantes extra-diagonales. Lorsque nous représenterons les composantes PDFs de ces tenseurs selon la théorie et les modèles, nous ne présenterons qu’un couple de composantes diagonale/extra-diagonale pour ne pas surcharger le corps.

Étude de la décomposition de Leonard des τ^{mod} - Les scores globaux sur les composantes L_{ij}^{ML} et L_{ij}^{Clark} sont très bons : les grandeurs FDNS et prédites par IA sont corrélées (au sens de Pearson) à 100 %, la variance manquante est de l’ordre de 1% et l’erreur relative est inférieure à 10%. Les mêmes quantités estimées par le modèles de Clark sont corrélées à 98 voire 99% et perdent au plus 5% de la variance. L’erreur relative est un peu plus élevée en moyenne sur les composantes de l’ordre de 15%.

Nous traçons les distributions des paires L_{yy}/L_{yz} figure Fig.(5.9) pour la théorie, la prédiction et les modèles. Au niveau des distributions, on constate que la distribution issue de l’IA épouse presque parfaitement celle de la FDNS pour les deux types de composantes. Les distributions des L_{ij} issues du modèle de Clark sont aussi très proches des distributions théoriques si ce n’est un décrochement au niveau des événements rares et plus énergétiques des composantes diagonales. Globalement, ces résultats montrent que les interactions faisant intervenir des échelles résolues sont bien modélisées. Le constat est très similaire pour les modélisations des composantes du tenseur C_{ij} . Les scores sont presque similaires que ceux obtenus sur les composantes de L_{ij} . Nous traçons les distributions figure Fig.(5.10). L’analyse des distributions est la même que précédemment, si les performances de l’IA ne dépendent pas du type de composante étudiée,

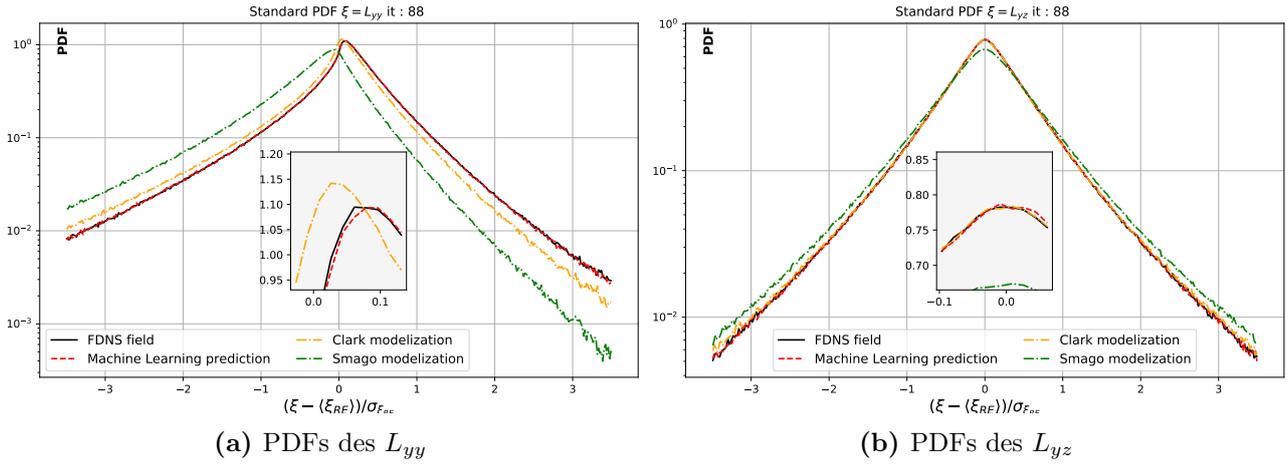


Figure 5.9 – À l’itération 88, comparaisons des PDFs de L_{yy} et L_{yz} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de L_{ij}^{FDNS} .

les modèles sont plus imprécis à propos des composantes diagonales avec un décrochage aux événements intermittents aux queues des distributions pour le modèle de Clark.

Études sur R_{ij} - Si les améliorations ne sont pas tellement visibles au niveau des tenseurs résultant des interactions entre échelles résolues et non résolues, la différence est plus marquée pour le tenseur R_{ij} représentant les contraintes issues des interactions des échelles non résolues entre elles ; ce sont les contraintes auxquelles les modèles LES sont intrinsèquement aveugles qui en constitue le principal défi. Les scores globaux pour les R_{ij}^{ML} et R_{ij}^{Clark} sont rapportés dans le tableau Tab.(5.6). Globalement, les scores sont assez bas, quand ils ne sont pas complètement décorrélés. Nous commençons par détailler les résultats du modèle de Clark. On voit que les scores Pearson sont négatifs pour toutes les composantes, synonymes de décorrélation. Il n’est

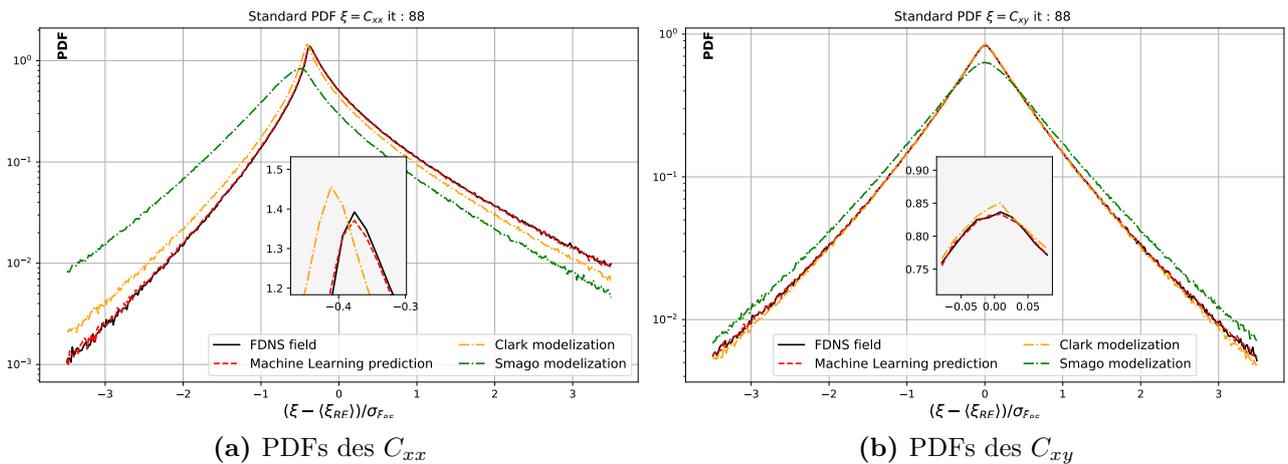


Figure 5.10 – À l’itération 88, comparaisons des PDFs de C_{xx} et C_{xy} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de C_{ij}^{FDNS} .

Scores	R_{xx}	R_{xy}	R_{xz}	R_{yy}	R_{yz}	R_{zz}
Pearson	0.78 / -0.71	0.57 / -0.52	0.56 / -0.52	0.79 / -0.72	0.56 / -0.51	0.78 / -0.71
$1 - \mathcal{R}^2$	0.47 / > 1	> 1 / > 1	> 1 / > 1	0.46 / > 1	> 1 / > 1	0.49 / > 1
E_r	0.64 / 1.57	1.1 / 1.59	1.1 / 1.6	0.63 / 1.56	1.1 / 1.57	0.65 / 1.57

Table 5.6 – Scores obtenus sur les composantes R_{ij} calculées selon la formule Eq.(2.4.3), pour l’itération 88. Les valeurs en noir représentent les scores sur R_{ij}^{ML} construites à partir de τ_{ij}^{ML} ; en orange, les composantes R_{ij}^{Clark} calculées à partir des τ_{ij}^{Clark} .

donc pas étonnant que les erreurs relatives soient au delà des 150% et que plus de 100 % de la variance soit perdue par la modélisation. En fait, les grandeurs construites par τ_{ij}^{Clark} n’ont plus rien à voir avec les grandeurs attendues. Les scores de l’IA sont meilleurs à tous les niveaux, mais ne sont pas aussi bons que ceux des composantes L_{ij}^{ML} et C_{ij}^{ML} . Les corrélations de Pearson sur les quantités prédites sont aux alentours de 80 % pour les grandeurs diagonales et à hauteur de 60% pour les composantes extra-diagonales. Le point noir du tableau est la variance qui est à 100% perdue pour les composantes extra-diagonales ainsi que les erreurs relatives au delà des 100%. Ces différents résultats sont malgré tout encourageants car ils prouvent que la prédiction contient plus d’information au niveau de la physique sous-maille à laquelle les modèles que nous avons considérés semblent être complètement aveugles. De façon quelque peu contre intuitive, les distributions ne reflètent pas ces grandes différences de précision (voir figure Fig.(5.11)); mais encore une fois rappelons que les PDFs ne considèrent pas la répartition des événements dans l’espace mais leur probabilité de survenir. Au niveau des

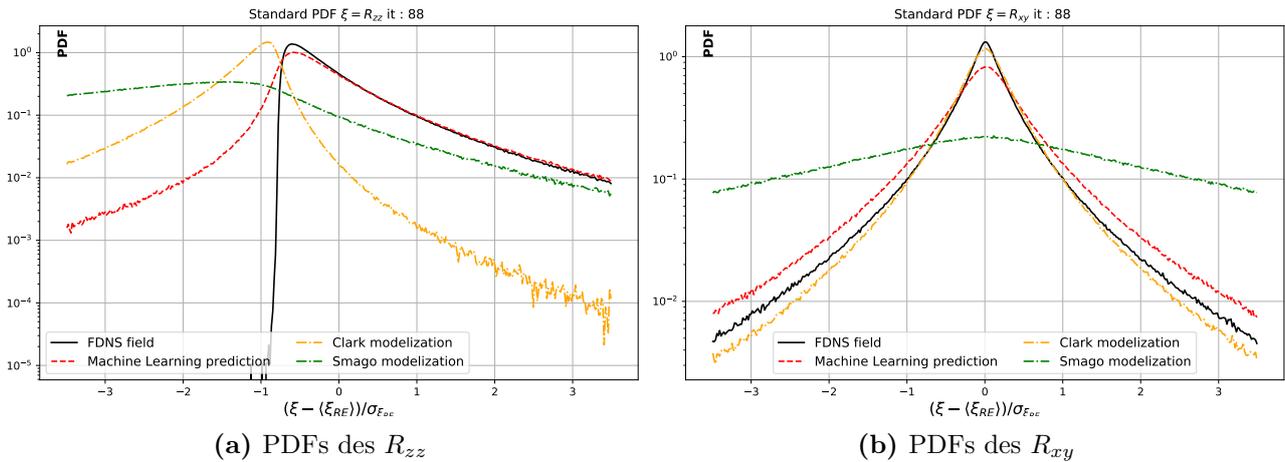


Figure 5.11 – À l’itération 88, comparaisons des PDFs de R_{zz} et R_{xy} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de R_{ij}^{FDNS} .

composantes diagonales, en combinant l’allure des PDFs et les mauvais scores de corrélations, il est clair que les modèles fournissent des tenseurs qui n’ont pas (pour Clark et Smagorinsky) ou peu (pour la prédiction) d’informations concernant les interactions sous mailles. Au niveau extra-diagonales, bien que les scores soient pires que pour les composantes diagonales, les distributions du modèle de Clark semblent plus cohérentes avec cette physique sous mailles. C’est ici évidemment contre intuitif et il est nécessaire de considérer les coupes des deux types de composantes comme précédemment. Nous analysons donc les coupes de ces composantes R_{zz} et R_{xy} théoriques, prédites et modélisées par le modèle de Clark. Nous effectuons ces coupes dans les figures respectives Figs.(5.12) et Figs.(5.13) effectuées dans le plan $z = 3$. Ces graphes peuvent expliquer plusieurs points. Dans un premier temps, on voit figures Fig.(5.11), dans les figures du haut, que la modélisation est effectivement insuffisante pour rendre compte des contraintes liées aux interactions sous-mailles. L’échelle de couleurs fixée sur celle de R_{zz}^{FDNS} , le champ R_{zz}^{Clark} est complètement bleu, c’est-à-dire nul voire négatif comme on le voit dans la figure du bas à droite qui montre les différence en valeur absolue entre champs théorique et modélisé. Sur la figure en haut au centre, on remarque que le champ prédit manifeste une topographie globale et les zones énergétiques attendues. Cependant, certaines zones sont “trop rouges” c’est-à-dire que la contrainte dans ces zones est surestimée comme supposé par la figure de la répartition spatiale des erreurs en valeurs absolues entre les champs théorique et prédit (figure du bas au centre). Toutefois, ces zones ne sont pas étendues et la magnitude des erreurs est deux fois plus faible que celles relevées dans la modélisation, de l’ordre des 80% à certains endroits et moins élevées ailleurs (expliquant les 63% d’erreurs relatives sur les composantes diagonales Tab.(5.6)).

Au niveau des composantes extra-diagonales, les figures Figs.(5.13) permettent également de constater les différences de performances entre prédiction et modélisation mais de façon plus subtile. Pour commencer, constatons que les gammes de valeurs sont différentes que précédemment, et qu’elles admettent des valeurs négatives. Le fait que les distributions extra-diagonales des grandeurs issues de la modélisation de Clark soient plus en cohérence avec les distributions théoriques est “un effet d’optique” puisque les amplitudes sont quasiment centrées autour de zéro et que l’on a déjà remarqué la tendance à sous estimer la composante R_{zz} . Ainsi, il semble que le modèle Clark ne contient que très peu d’information sur \mathbf{R} . Concernant la prédiction, le constat est différent et beaucoup plus positif à commencer par les corrélations qui étaient d’emblée plus cohérentes. Les zones de contraintes négatives sont bien représentées dans l’ensemble, ainsi que les zones de l’extrême opposé. On remarque des erreurs en valeurs absolues éparpillées avec des pics à 100 % mais malgré tout sporadiques. Topographiquement, la coupe de la prédiction ressemble très fortement à celle de la FDNS. Remarquons la présence, inédite jusqu’ici, de “la trace” de la discrétisation en cubes (dans le champ R_{xy}^{ML} . Cette trace pourrait indiquer que les erreurs relevées sur les coupes du tenseur τ_{ij}^{ML} figure Fig.(5.5f) en bas au centre, impactent \mathbf{R}^{ML} .

Étude au niveau dissipatif : apport de l’IA - De la même façon qu’il est possible d’établir le taux de dissipation de l’énergie cinétique turbulente par les interactions impliquant les échelles non résolues $\Pi = -\bar{S}_{ij}\tau_{ij}$, on peut essayer de voir comment se répercutent les erreurs de prédiction/modélisation au niveau dissipatif en décomposant τ_{ij} comme précédemment. Il est alors possible de construire Π_{Lij}^{Mod} , Π_{Cij}^{Mod} et Π_{Rij}^{Mod} que nous avons déjà définis équations

Scores	Π_{Lij}	Π_{Cij}	Π_{Rij}
Pearson	0.99 / 0.99	0.99 / 0.99	0.6 / -0.58
$1 - \mathcal{R}^2$	0.01 / 0.03	0.02 / 0.04	> 1 / > 1
E_r	0.11 / 0.17	0.11 / 0.17	1.1 / 1.7

Table 5.7 – Scores obtenus sur les différentes dissipations reliées aux termes de la décomposition de Leonard pour l’itération 88. En noir, les scores évalués sur les grandeurs issues de τ_{ij}^{ML} ; en orange, les scores calculés en considérant les grandeurs issues de τ_{ij}^{Clark} .

Eqs.(2.4.5), (2.4.6) et (2.4.7), respectivement. Ici aussi la décomposition n’est pas exacte et

$$\Pi_{\tau_{ij}}^{Mod} \neq \Pi_{L_{ij}}^{Mod} + \Pi_{C_{ij}}^{Mod} + \Pi_{R_{ij}}^{Mod}$$

nous estimons plutôt l’écart entre chaque tenseur ainsi construit et la théorie, de façon indépendante les unes des autres. Comme précédemment, le modèle augmenté et de Clark estiment une dissipation dues aux interactions décrites par L_{ij} très corrélée avec la contribution cible L_{ij}^{FDNS} . Idem au niveau des interactions décrites par C_{ij} . Ces résultats sont attendus au vue des excellents scores obtenus sur ces deux tenseurs. Sans surprise également, la dissipation causée par

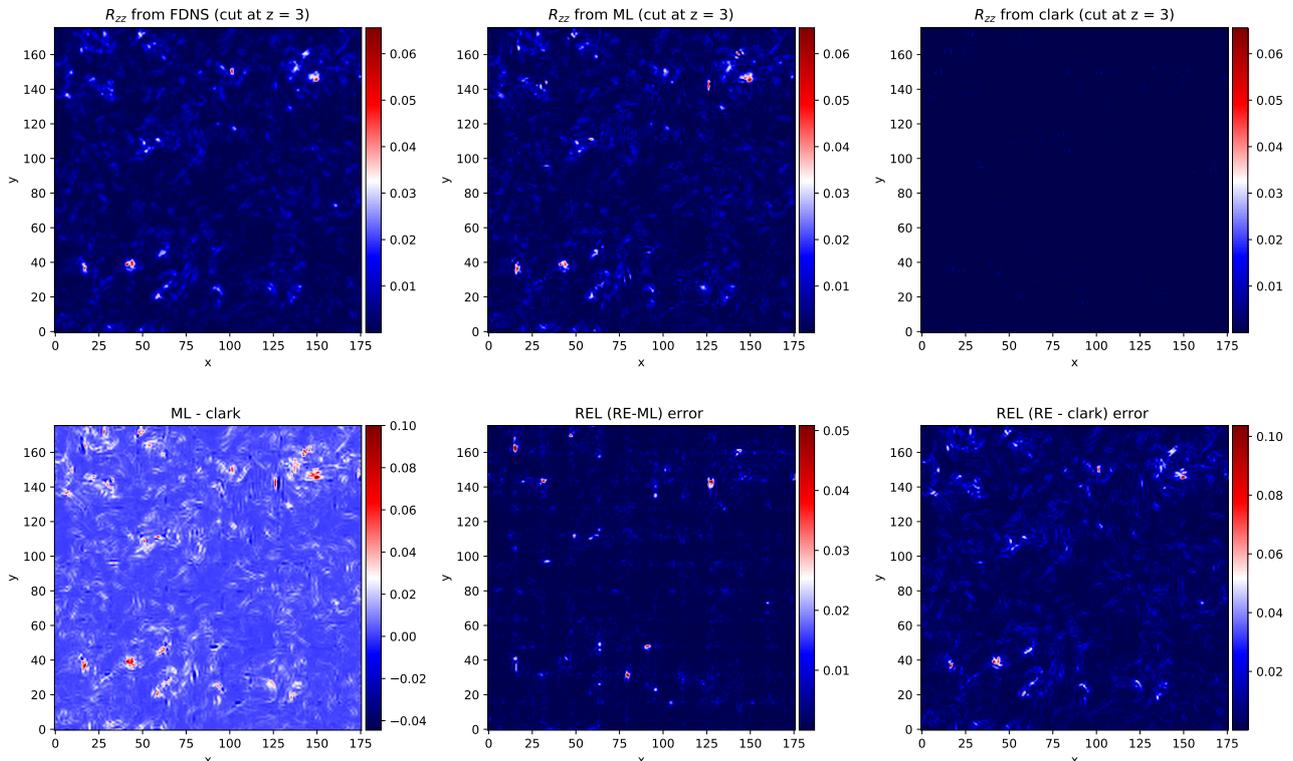


Figure 5.12 – **Figures du haut** : Coupe de la composante R_{zz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 3$.

les interactions des échelles non résolues entre elles n'est pas aussi précise. Nous renseignons les scores des trois dissipations d'origines différentes dans le tableau Tab.(5.7). Nous traçons également les coupes dans l'espace physique des grandeurs $\Pi_{R_{ij}}^{\text{FDNS}}$, $\Pi_{R_{ij}}^{\text{ML}}$ et $\Pi_{R_{ij}}^{\text{Clark}}$ Fig.(5.14). Globalement, on remarque que les figures du haut sont assez proches, précisons que les échelles de couleurs sont fixées selon les extrema du champ $\Pi_{R_{ij}}^{\text{FDNS}}$ (figure du haut à gauche). Cependant, quand on analyse plus profondément la modélisation (figure du haut à droite), on remarque que les zones d'activités ont des amplitudes opposées par rapport à la FDNS. C'est la raison pour laquelle la répartition spatiale de l'erreur fait apparaître des zones d'erreur (figure du bas à gauche) néanmoins des pics autour de 200 %. Le champ prédit (en haut au centre) présente une topologie encore plus proche de celle du champ FDNS, avec des erreurs (en bas au centre) dépassant rarement les 100%, sans inversions. Les zones de concentration d'erreurs sont également moins nombreuses que pour la modélisation. Ces constatations sont suffisantes pour prouver la qualité supérieure de la prédiction par rapport à la modélisation, informant que non seulement le tenseur $\boldsymbol{\tau}$ est mieux modélisé par la prédiction mais il est également plus riche en Physique et pourrait permettre une élaboration de modèles plus *sensés*.

En conclusion, dans le régime développé de la turbulence, la prédiction du CNN surpasse le modèle de Clark aux niveau tensoriel, vectoriel et scalaire. De plus, elle permet de recouvrer plus de Physique que ce dernier modèle. A priori, sur ce régime, dans la gamme $Re_\lambda \in [90 ; 140]$, un modèle construit par intelligence artificielle et entraîné comme nous l'avons développé dans

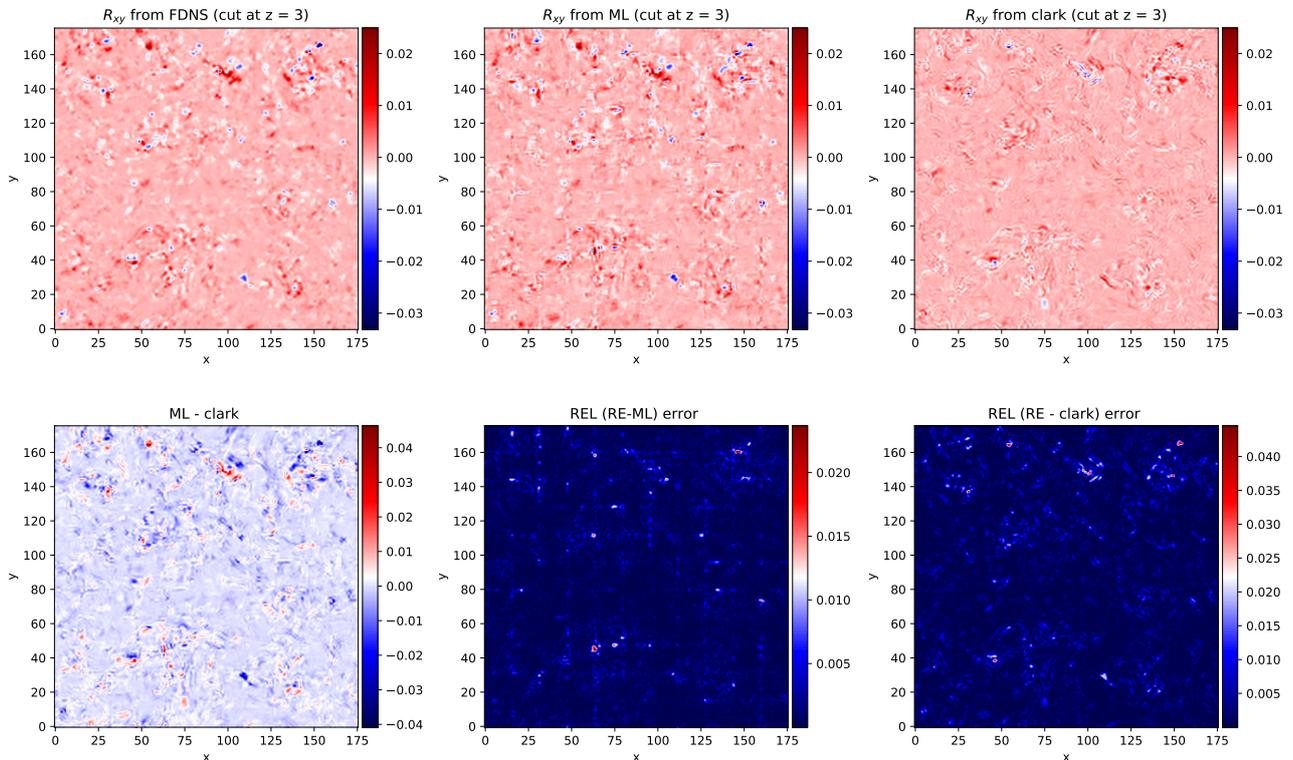


Figure 5.13 – **Figures du haut** : Coupe de la composante R_{xy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 3$.

les chapitres précédents, est capable de prédire des résultats quantitativement mais également qualitativement plus viables puisqu'ils modélisent véritablement les interactions sous-mailles avec une précision de plus de 70% en moyenne.

Nous testons à présent notre modèle sur des champs de vitesse complètement différents en nous plaçant à l'itération 42 (voir figure Fig.(5.1)).

5.1.3 Capacité d'adaptation par rapport à des champs turbulents de nature fondamentalement différente

Nous débutons notre analyse en considérant les scores de performance globaux de la prédiction en elle même (le niveau tensoriel) ; ils sont présentés dans le tableau Tab.(5.8). Il est intéressant d'en comparer les valeurs avec celles du tableau de l'itération 118 Tab.(5.2) ; la similitude des résultats permet d'établir les mêmes conclusions que précédemment section 5.1.1 à savoir : par rapport au modèle de Clark, le modèle augmenté couvre mieux, et de manière plus riche³, les six composantes du tenseur sous mailles. Au sens de ces différentes métriques, la modélisation

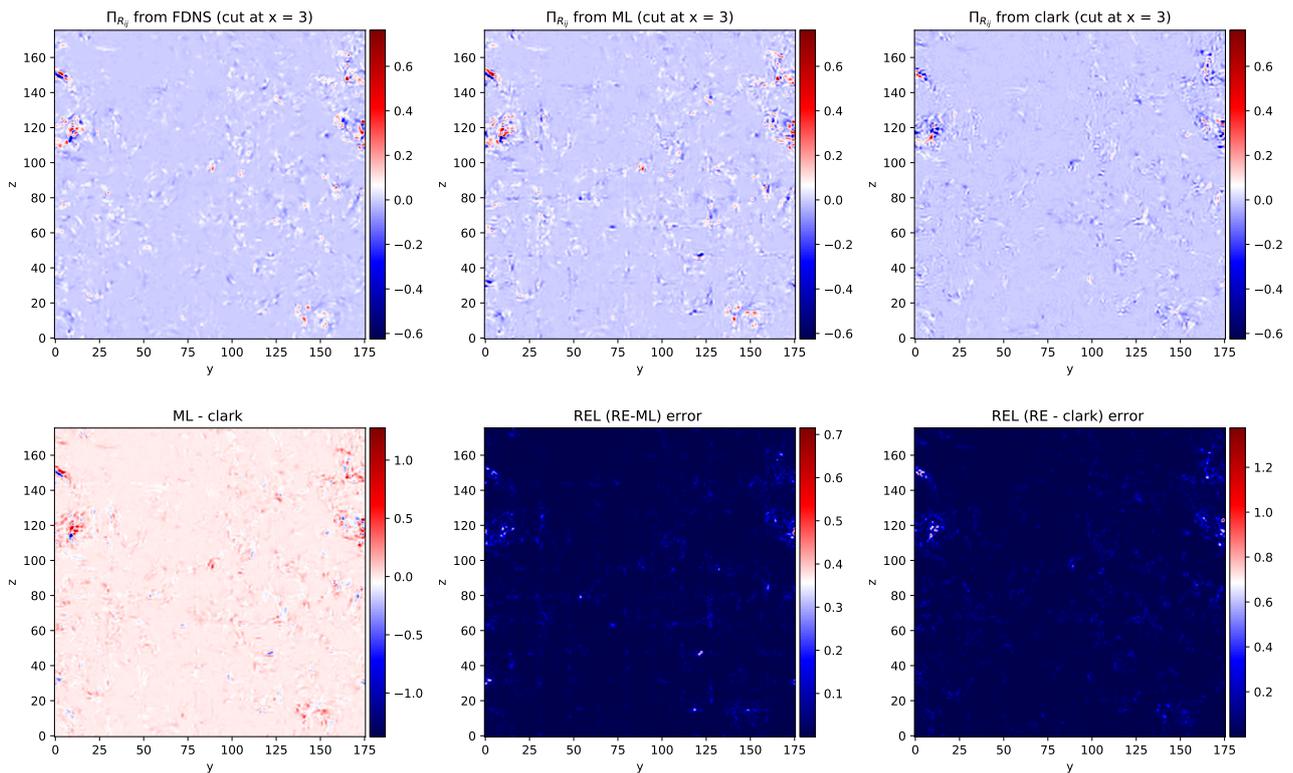


Figure 5.14 – Figures du haut : Coupe de la composante R_{xy}^{FDNS} (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$. Les échelles de couleurs sont celles de la FDNS.

3. Au sens de l'information recouvrée sur la variance mesurée par le score \mathcal{R}^2

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.99 / 0.97					
$1 - \mathcal{R}^2$	0.02 / 0.10	0.03 / 0.08	0.03 / 0.07	0.02 / 0.10	0.03 / 0.07	0.02 / 0.10
E_r	0.11 / 0.28	0.18 / 0.27	0.18 / 0.27	0.11 / 0.28	0.18 / 0.27	0.12 / 0.28

Table 5.8 – Scores obtenus sur la qualité de la prédiction au niveau tensoriel pour l’itération 42. Les valeurs en noir représentent les scores sur les prédictions de l’IA. En orange, nous renseignons à titre de comparaison la qualité de la modélisation par le modèle Clark des composantes τ_{ij} .

par IA est très proche des résultats théoriques attendus, fermant ainsi les équations de Navier-Stokes dans le cadre de la LES.

Au niveau de l’impact de la prédiction sur la quantité de mouvement ($\nabla \cdot \boldsymbol{\tau}$) ainsi qu’au niveau de la dissipation Π et de la diffusion de l’énergie par interactions entre échelles résolues et non résolues Ψ , nous renseignons les scores tableau Tab.(5.9).

Ces composantes mesurent directement ou indirectement la faisabilité d’un modèle uniquement construit par IA, au delà de sa performance à modéliser les interactions sous mailles. Le constat général est le suivant : les erreurs de prédictions prennent légèrement plus d’ampleur au travers les contractions tensorielles ; cependant, les imprécisions mesurées sur les grandeurs construites à partir de la prédiction τ_{ij}^{ML} demeurent plus faibles que celle mesurées sur leurs homologues basées sur τ_{ij}^{Clark} . Ces constatations étaient également établies plus haut.

Au niveau des PDFs, nous nous attendons aussi à des résultats similaires à ceux présentés dans la section 5.1.1. Pour ne pas surcharger le corps, nous ne tracerons que les PDFs des niveaux vectoriel et scalaire ; les figures équivalentes pour le niveau tensoriel sont reportées dans l’annexe

Scores	$\{\nabla \cdot \boldsymbol{\tau}\}_x$	$\{\nabla \cdot \boldsymbol{\tau}\}_y$	$\{\nabla \cdot \boldsymbol{\tau}\}_z$	Π	$u \cdot (\nabla \cdot \boldsymbol{\tau})$
Pearson	0.93 / 0.86	0.93 / 0.86	0.92 / 0.84	0.97 / 0.92	0.93 / 0.87
$1 - R^2$	0.15 / 0.27	0.14 / 0.27	0.15 / 0.27	0.06 / 0.15	0.14 / 0.26
E_r	0.39 / 0.52	0.38 / 0.52	0.38 / 0.52	0.24 / 0.37	0.37 / 0.50

Table 5.9 – Scores obtenus aux niveaux vectoriel et scalaire pour l’itération 44. Les valeurs en noir représentent les scores obtenus par les prédictions de l’IA. En orange, nous renseignons les scores globaux pour le modèle Clark des composantes $\partial_{x_j} \tau_{ij}$ et les grandeurs scalaires Π et $u_i \partial_{x_j} \tau_{ij}$.

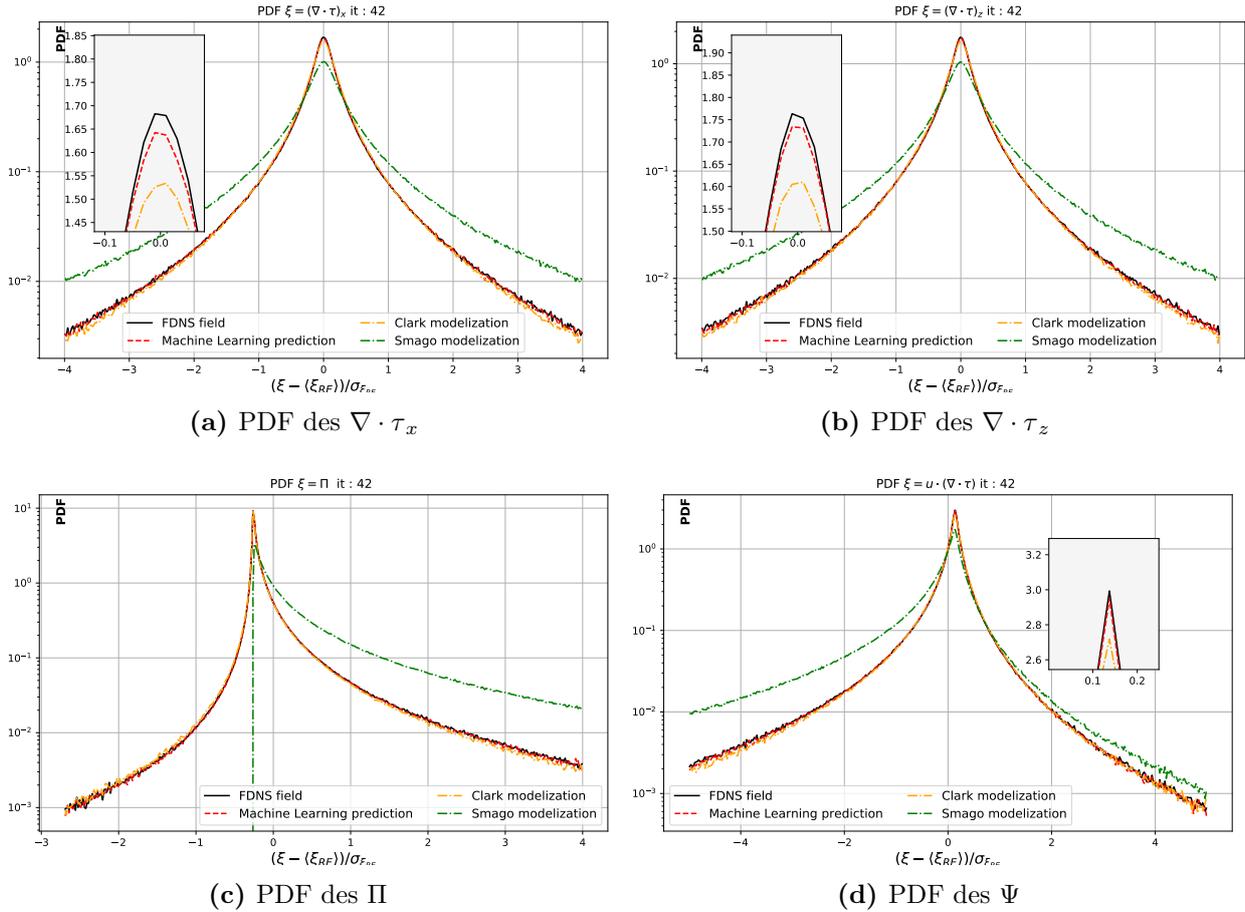


Figure 5.15 – À l’itération 42, comparaisons des PDFs des composantes x et z de $\partial_{x_j} \tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenues à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte); Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

figure Fig.(B.4). Comme précédemment, nous voulons comparer la distribution de probabilités d’obtenir certaines valeurs à partir des prédictions avec les distributions théoriques attendues. Ici aussi, nous renseignons à titre informatif les PDFs des niveaux vectoriel et scalaire obtenues à partir des modélisations de Clark et de Smagorinsky. Nous traçons ces PDFs pour deux composantes de la divergence de τ à savoir les composantes selon \vec{x} et \vec{z} , et nous traçons les PDFs des deux grandeurs scalaires Π et Ψ . Les PDFs des différentes grandeurs considérées construites théoriquement, à partir de l’IA ou du modèle de Clark sont très similaires. L’intermittence est toujours mieux retrouvée dans les distributions issues des prédictions que celles construites à partir du modèle algébrique. En revanche la différence est moins marquée que lors des itérations précédentes supposant que les erreurs du modèle Clark sont moins fortes loin du régime développé. Il semblerait donc qu’un modèle algébrique nécessiterait des informations supplémentaires aux gradients pour une modélisation plus fiable de la physique à ce régime.

Les scores des prédictions présentées à ces itérations - choisies pour englober les régimes de la turbulence en amont, en aval du pic de dissipation et après la stabilisation de cette grandeur et de l’énergie cinétique - sont très encourageants. Les propriétés de généralisation testées avec

succès reflètent l’efficacité de l’entraînement de l’IA, de l’optimisation de son architecture et de sa complexité et permettent les très bonnes performances concernant la prédiction du tenseur de contraintes sous mailles dans des écoulements de nature plus turbulente et à même d’explicitier plus de complexités et de richesses tant aux niveaux des structures que de leurs interactions. Nous résumons finalement l’évolution des corrélations de Pearson en fonction du temps dans les figures Figs.(5.16), (5.17). Ces figures permettent de mieux appréhender les conclusions établies et celles qui vont suivre, et complètent l’analyse des capacités de l’IA - sélectionnée (section 4.3), et entraînée (section 4.4) sur une simulation turbulente précise - à prédire les six composantes du tenseur des contraintes à partir d’un champ de vitesse filtré issu d’une simulation de nature différente, plus turbulente. D’autres itérations (comme l’itération 25 voir l’annexe B.1.4 pour plus de détails) sont considérées dans ces figures pour étendre nos conclusions ou les limiter. Ces graphiques sont aussi l’occasion de présenter des corrélations sur les grandeurs L_{ij} , C_{ij} et R_{ij} à d’autres itérations.

L’approche qui a été la notre se base sur une analyse pragmatique des capacités présumées d’une intelligence artificielle. Une vision assez pessimiste tend à conjecturer que plus les champs de vitesse inédits sont proches de ceux constituant l’ensemble d’entraînement, plus les prédictions sont meilleures, voire exactes⁴ et inversement. Nos études ont montré que pour une simulation plus turbulente et complètement inédite, les capacités de l’IA s’avèrent être excellentes quelle que soit la composante du tenseur τ que l’on étudie. Il a été également souligné que le découpage en cubes impliquait une baisse de la précision sur les points constituant les faces de ces cubes⁵. Cependant cette baisse ne se fait pas véritablement ressentir sur les scores globaux puisque les composantes τ_{ij}^{ML} sont à 99% corrélés au sens de Pearson avec τ_{ij}^{FDNS} , sur les itérations considérées. Les conclusions furent sensiblement les mêmes lors de l’analyse des distributions de valeurs prédites majoritairement identiques aux distributions des composantes de τ_{ij}^{ML} . Nous avons jugé nécessaire d’établir que l’analyse des PDFs **doit** être couplée aux scores des corrélations, car les premières ne tiennent pas compte de la représentation spatiale des valeurs qu’elle quantifie et débouche sur des “effets d’optique” ; la résolution du paradoxe de fortes similarités des distributions pour des modélisations faiblement corrélées avec les champs théoriques s’est faite en considérant plusieurs coupes des champs *physiques* concernés.

Nous l’avons souvent mentionné, notre analyse ne s’arrête pas à la prédiction du réseau en soi ; elle s’étend également aux dérivées de ce tenseur, ainsi qu’aux interactions de ce champ prédit avec ceux disponibles dans la LES comme le tenseur de déformation $\overline{S_{ij}}$. Les études à ces niveaux permettent de surveiller la diffusion des effets des interactions sous-mailles ainsi que leur magnitude, capitales pour calculer les champs de vitesse aux itérations successives ; mais également le devenir de l’énergie cinétique en évaluant si la dissipation est sur/sous estimée évoquant alors une corruption au niveau des échanges énergétiques entre les structures etc. Nous avons prouvé que les contractions basées sur τ_{ij}^{ML} sont très corrélées avec celles impliquant τ_{ij}^{FDNS} ; dans ce cas les scores sont plus élevés que ceux mesurant la corrélation entre ces grandeurs construites à partir de τ_{ij}^{Clark} et leurs homologues issues de la FDNS.

Enfin, nous avons tenté de comprendre qualitativement la supériorité des prédictions de l’IA par rapport à la modélisation de Clark. Nous avons proposé une façon de mesurer l’écart sur chaque terme de la décomposition de Leonard construit à partir de τ_{ij}^{ML} et τ_{ij}^{Clark} . Les conclu-

4. Au sens des différentes métriques introduites section 3.3 et 2.4.2

5. Voir à ce propos les figures de corrélations le long des axes Fig.(5.4) ou les comparaisons de champs eux mêmes comme Fig.(5.6) ou Fig.(5.8), Figs.(5.10) etc.

sions établies section (5.1.2) sont étendues aux autres itérations considérées Fig.(5.17). D'une part, les interactions entre échelles résolues et non résolues regroupées dans les termes L_{ij} et C_{ij} sont presque parfaitement modélisées par l'IA et par le modèle de Clark Figs.(5.17a) et (5.17b). D'autre part, le modèle du Gradient apparaît comme aveugle aux interactions entre échelles non résolues au vue des corrélations négatives avec les grandeurs R_{ij}^{FDNS} , alors que cette physique est à 60% en moyenne retrouvée par le tenseur τ_{ij}^{ML} (corrélations atteignant les 80% pour les composantes diagonales des R_{ij} prédits et cibles) voir figure Fig.(5.17c). Notons également, que l'inversion des valeurs constatées dans $\mathbf{R}^{\text{Clark}}$ peut expliquer sa tendance à sur-évaluer la rétro diffusion qui le caractérise (Vreman et al., 1996).

Nous voulons étendre les conclusions établies pour une simulation encore plus turbulente, tant aux différents niveaux d'analyse que nous nous sommes fixés, qu'au niveau des composantes de Leonard.

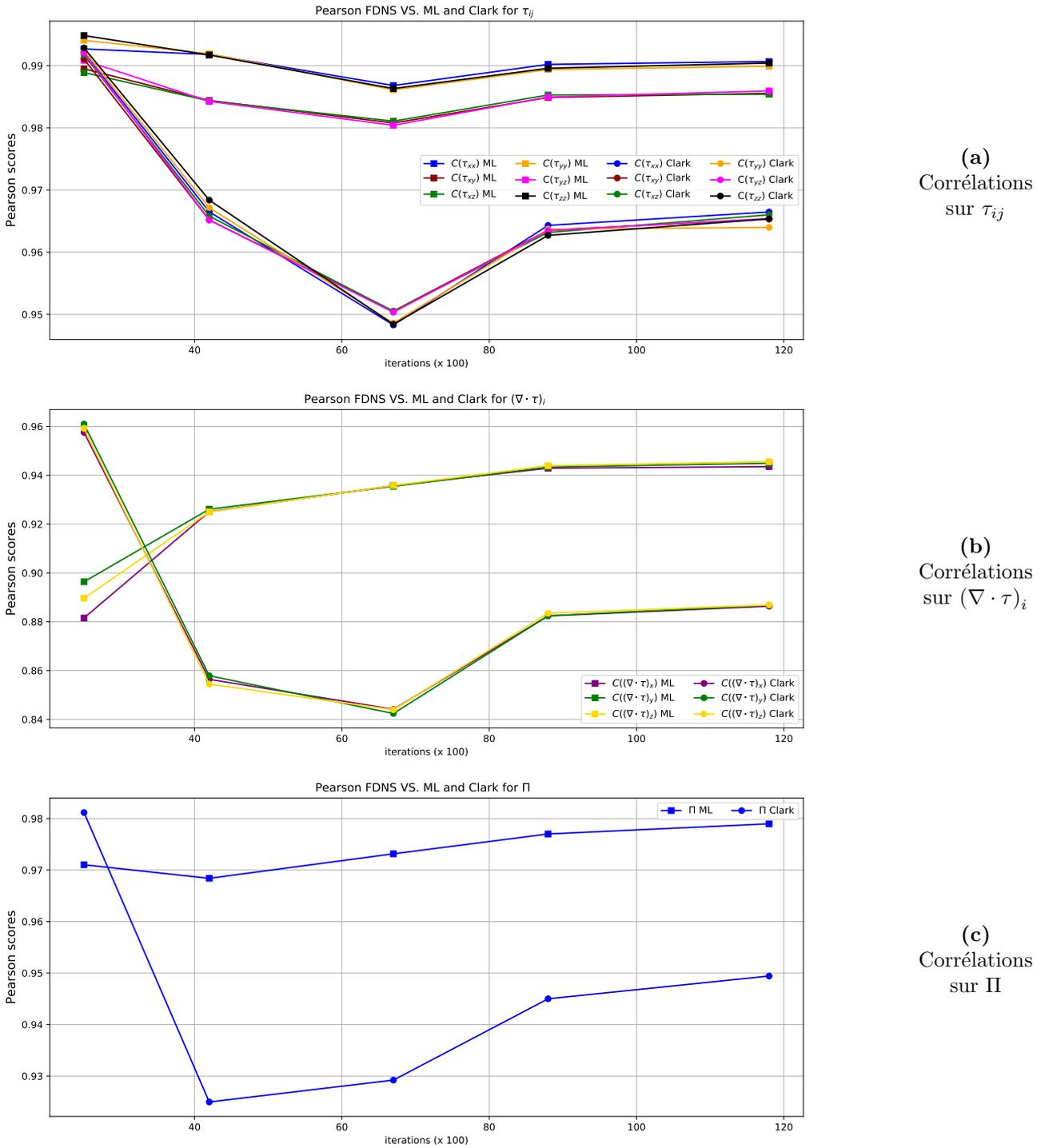


Figure 5.16 – Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) aux niveaux tensoriel, vectoriel et scalaire.

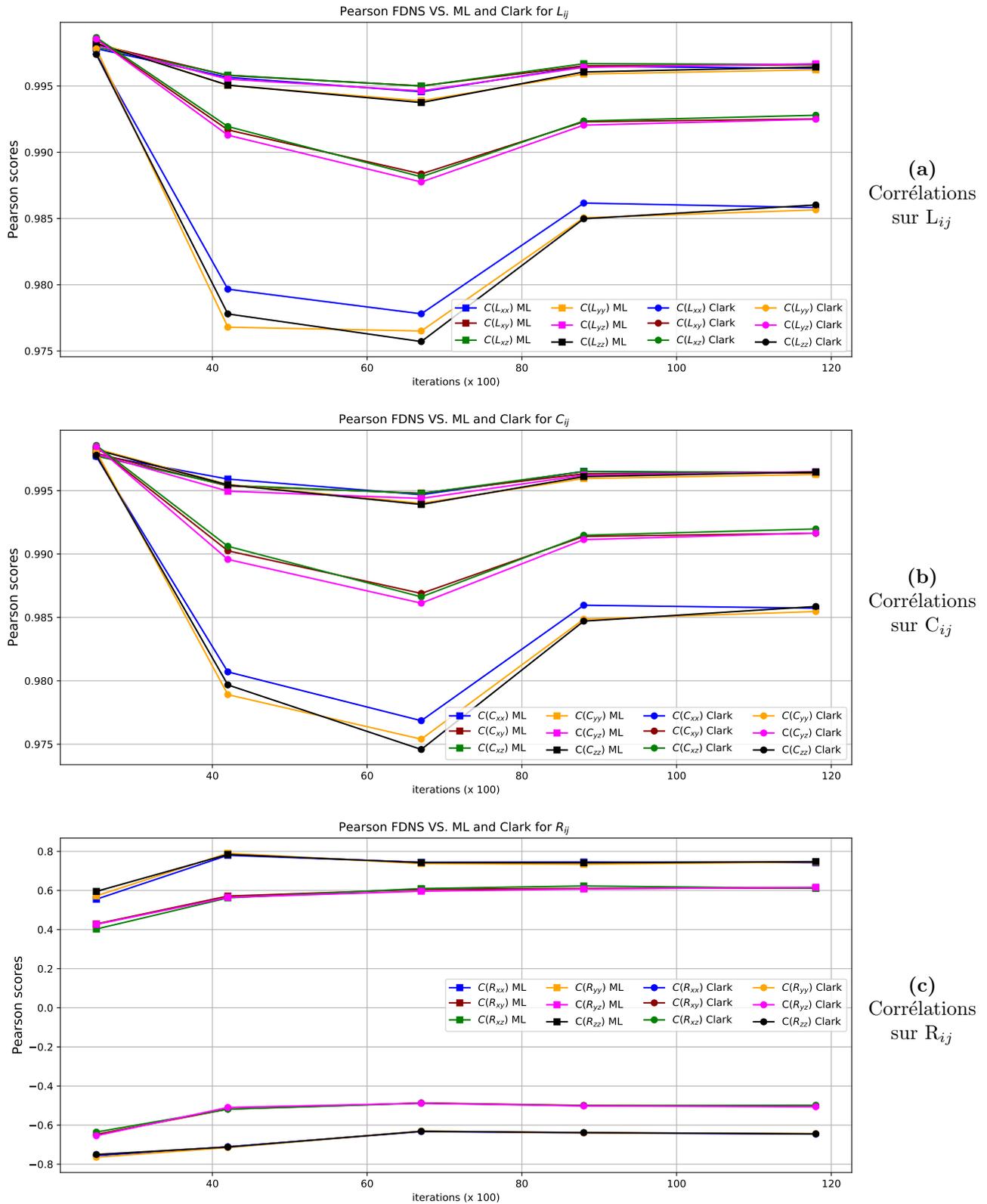


Figure 5.17 – Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) des trois types d’interactions composant le tenseur τ conformément à la décomposition de Leonard (2.3.6).

5.2 Prédications sur une simulation avec $Re_\lambda \approx 240$

Les résultats sur la simulation précédente sont très encourageants. À présent, nous voulons déterminer si la relation établie entre champ de vitesse et contraintes dues aux échelles non résolues est valable et généralisable. Comme précédemment, nous analyserons les prédictions sur trois itérations représentant trois états de la turbulence. Ces trois points sont représentés par des étoiles vertes sur les courbes des évolutions temporelles des grandeurs turbulentes, figure Fig.(5.18). Le spectre en fin de simulation est tracé figure Fig.(5.19). Nous illustrons les différences entre les champs de vitesse considérés lors de l'entraînement et ceux qui le seront dans cette section figure Fig.(5.20).

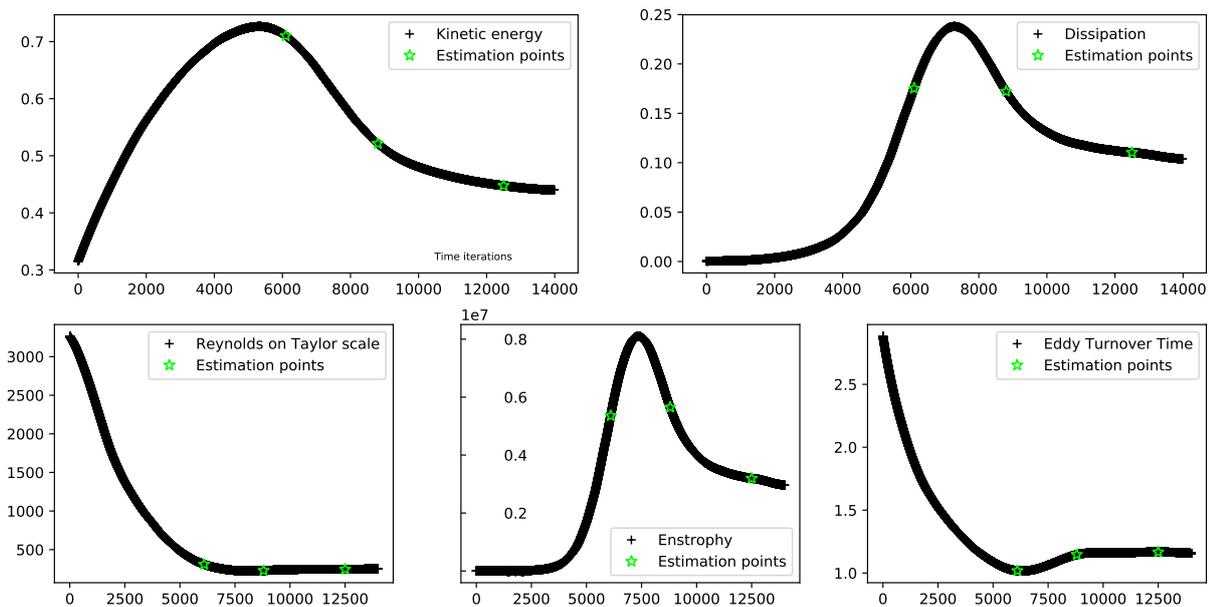


Figure 5.18 – Évolution temporelle des grandeurs caractéristiques de la Turbulence pour la simulation dont la résolution est $N = 256$ et $Re_\lambda \approx 240$ dans le régime développé.

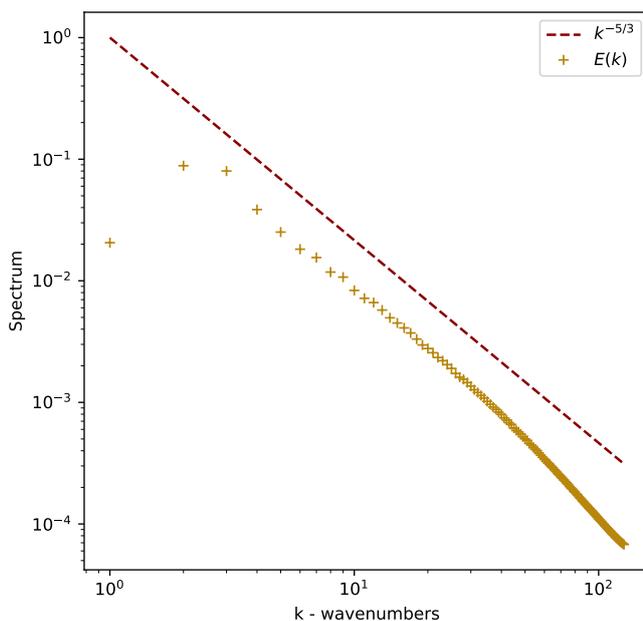


Figure 5.19 – Spectre de l'énergie cinétique calculée dans le régime développé de la turbulence au sein de l'écoulement issue de la simulation au Reynolds $Re_\lambda \approx 240$.

Contrairement à la simulation précédente, il n'existe qu'un seul point pour l'étude du régime développé de la turbulence, écoulement après la zone transitoire marquée par un pic en énergie cinétique suivi d'un pic en dissipation de cette même énergie (voir figure Fig.(5.18). Soulignons ici : concernant les tests considérés dans la simulation de la section précédente, les points sélectionnés étaient alignés sur une seule valeur de dissipation, correspondant à une activité des échelles dissipatives identique à différentes étapes de la turbulence⁶. Dans cette section, nous considérons les deux premiers points (61 et 88) pour lesquels la dissipation est identique, ils sont toutefois situés plus proches du pic de cette grandeur que précédemment. Le régime développé de la turbulence sera étudié à une activité des échelles sous mailles plus faibles, proche de la

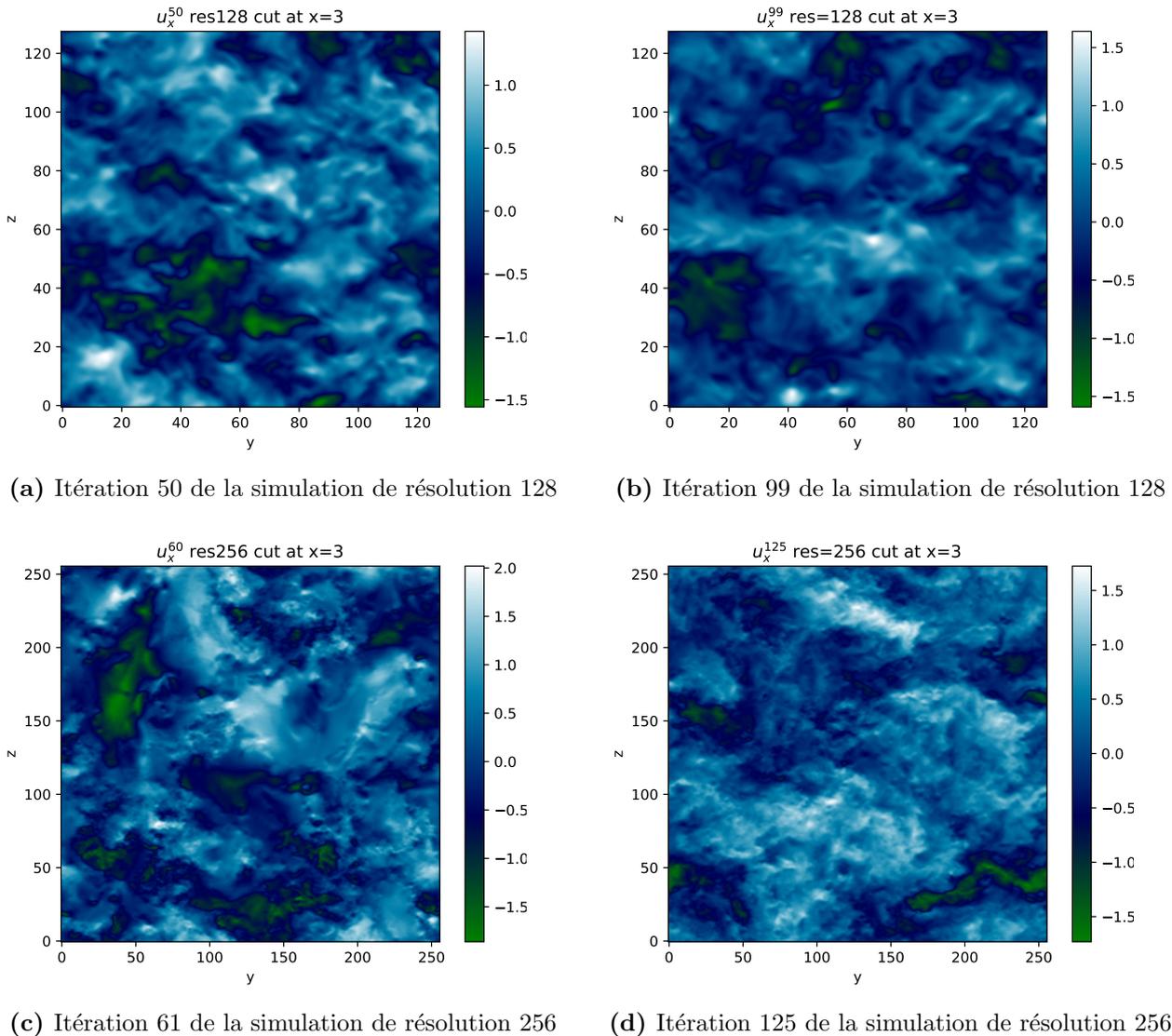


Figure 5.20 – Illustrations des premiers et derniers u_x balayées lors de l'apprentissage (figures du haut) et lors des tests de cette section de (figures du bas) Voir tableau Tab.(5.1).

6. L'activité est identique mais la situation de l'écoulement est radicalement différente d'une itération à l'autre : le premier point se situait avant la dissipation du trop plein d'énergie cinétique, un deuxième après la sur-dissipation et un troisième à l'équilibre, voir figure Fig.(5.1).

valeur asymptotique.

Une augmentation du Reynolds est corrélée à une augmentation de la résolution. Aussi, pour une taille de domaine global identique ($2\pi \times 2\pi \times 2\pi$), nous considérerons 256 points de discrétisation dans chacune des directions de l'espace, nécessitant 4096 cubes pour la reconstruction d'un seul champ. Enfin, dans tous les résultats qui vont suivre, le facteur R est maintenu à sa valeur $R = 2$ (voir discussion section 4.2.1 et notamment tableau Tab.(4.1)) imposant un nombre d'onde de coupure $\kappa_c = 64$. Il est intéressant alors de comparer les champs considérés lors de l'entraînement et ceux des itérations les plus extrêmes considérés pour cette nouvelle simulation.

5.2.1 Analyse des performances de la prédiction dans un écoulement plus turbulent dans son régime développé

Nous débutons à présent notre analyse sur la prédiction des composantes τ_{ij}^{ML} en turbulence développée ; c'est le niveau tensoriel. Nous considérons les trois scores globaux dans le tableau Tab.(5.10) ainsi que les erreurs relatives (en pourcentage) le long des différents axes figures Fig.(5.22). Ces dernières figures répondent aux problématiques que nous avons déjà abordées :

- Confirmer que l'isotropie constatée dans l'apprentissage ainsi que dans la généralisation aux champs de la simulation au $Re_\lambda = 140$ se retrouve au niveau des prédictions sur des champs de vitesse très différents
- Vérifier la présence et l'impact de la moindre performance au niveau des points aux bordures des cubes.
- De manière plus générale, établir si la relation apprise sur la base de données d'apprentissage est universelle.

Nous utilisons les mêmes notations et couleurs que dans les tableaux précédents : les valeurs noires représentent les scores des corrélations entre prédictions et données cibles, alors que les valeurs oranges représentent les scores sur les corrélations comparant modélisation par le modèle de Clark et les données cibles.

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.98 / 0.92	0.97 / 0.92	0.97 / 0.92	0.98 / 0.92	0.97 / 0.92	0.98 / 0.92
$1 - \mathcal{R}^2$	0.04 / 0.33	0.05 / 0.16	0.05 / 0.16	0.04 / 0.34	0.05 / 0.16	0.04 / 0.34
E_r	0.15 / 0.43	0.23 / 0.40	0.23 / 0.40	0.15 / 0.43	0.23 / 0.40	0.15 / 0.43

Table 5.10 – Scores au niveau tensoriel à l'itération 125. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark.

Globalement, les prédictions en volume τ_{ij}^{ML} effectuées à partir de champs de vitesse dans le régime développé sont très corrélées avec les composantes du tenseur cible τ_{ij}^{FDNS} , d’une part, et le sont bien plus que ne le sont les modélisations τ_{ij}^{Clark} . Les prédictions recouvrent près de 95% de la variance pour chacune des composantes et ont un écart relatif oscillant entre 15% et 25% pour les composantes diagonales et extra-diagonales respectivement. L’écart relatif entre les modélisations de Clark et les données FDNS est supérieur 40%; au deuxième ordre, les composantes τ_{ii}^{Clark} ne recouvrent que 70% de la variance.

La qualité des prédictions à cette itération est du même acabit que celle constatée sur le régime développé de la simulation précédente Tab.(5.2). Cette constatation n’est pas transposable sur la modélisation, comme on peut le voir notamment au niveau de la variance et de l’erreur relative qui sont plus fortes dans cette simulation. Ce résultat est de prime importance car il permet de confirmer que l’IA entraînée de façon la plus minimale⁷ respecte l’auto-similarité des écoulements turbulents développés, supposée par Kolmogorov. C’est-à-dire que du point de vue du réseau, les champs de vitesse en turbulence développée renferme l’information nécessaire et universelle pour modéliser les actions des échelles de la fin de la gamme inertielle et des échelles dissipatives. Cet argument fait aussi écho à la conservation nécessaire du facteur R entre entraînement et prédiction puisqu’il assure une conservation de la même quantité d’information.

Dans la continuité de notre approche, nous étudions les distributions de valeurs des composantes des tenseurs théoriques, prédits et modélisés. Rappelons que fondamentalement, les PDFs permettent de visualiser la fréquence des événements au sein des données, elle renseigne entre autres sur les événements plus rares presque négligés lors des calculs de corrélations globaux; nous avons donc accès à l’intermittence. Le comparatif des PDFs du niveau tensoriel est consigné figures Figs.(5.21). Ces figures mettent en exergue un résultat rencontré précédemment : les modèles fournissent des composantes extra-diagonales dont les valeurs, et leurs distribution sont assez similaires avec les valeurs attendues. En revanche, les distributions des valeurs observées dans les composantes diagonales sont différentes tant aux niveaux de l’intermittence que des événements courants. Cela a été en partie expliqué par le fait que les composantes diagonales véhiculent non seulement plus d’énergie que leurs homologues extra-diagonales, mais contiennent aussi plus de gammes de valeurs. Ainsi la justesse des modélisations est mieux challengée sur ces composantes. Les prédictions, quant à elle, exhibent des distributions très similaires aux distributions théoriques sans distinction de composantes, et sans distinctions de fréquence des événements. Cette différence, couplée avec les scores penchant également en faveur de la prédiction, suppose à présent que les modèles ne disposent pas assez d’informations pour décrire l’ensemble des gammes de valeurs possibles. Bien sûr, nous vérifierons cette supposition aux autres itérations.

Rappelons encore une fois, de prime abord les PDFs issues du modèle Smagorinsky sont plus proches des distributions FDNS que celles issues du modèle de Clark. Or, les corrélations des composantes diagonales avec celles de la FDNS sont de l’ordre de 20% et celles sur les composantes extra-diagonales de l’ordre de 30%. Cette apparente contradiction est un autre exemple de l’“effet d’optique”. Nous résolvons cette incohérence par le fait que les distributions ne donnent aucune information sur la localisation spatiale des grandeurs, ni les gammes que **le modèle** peut exhiber. Nous traçons dans l’annexe les coupes de τ_{yz}^{FDNS} , τ_{yz}^{ML} et τ_{yz}^{Smago} ainsi que les erreurs en valeur absolue associées à chaque modélisation.

7. En terme de diversité des simulations considérées lors de son apprentissage

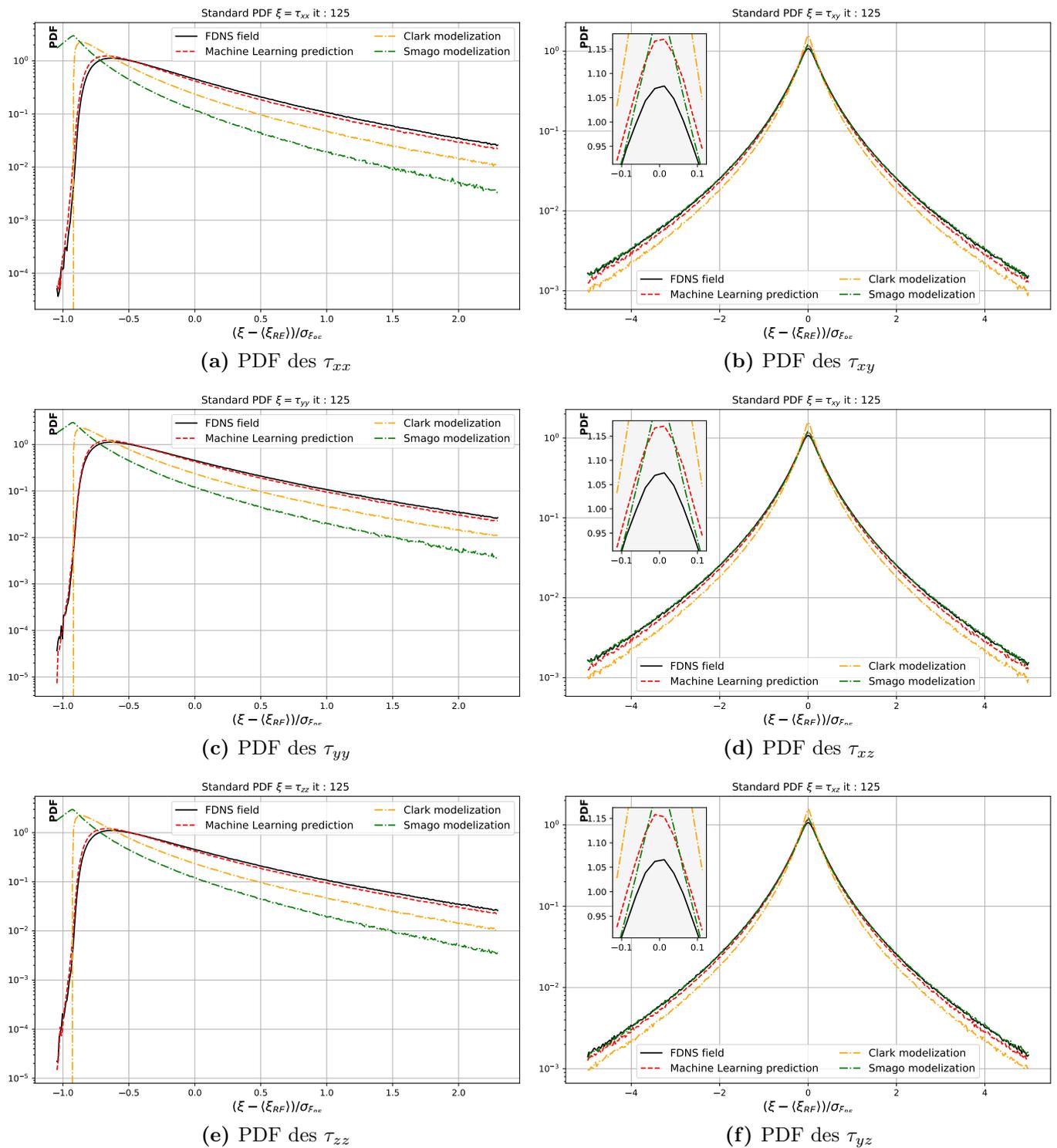


Figure 5.21 – À l'itération 125, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

Les autres niveaux sont indispensables puisqu'ils permettent de savoir si un éventuel modèle

Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.92 / 0.77	0.92 / 0.76	0.92 / 0.77	0.97 / 0.90	0.92 / 0.77
$1 - R^2$	0.15 / 0.42	0.15 / 0.42	0.15 / 0.42	0.06 / 0.23	0.15 / 0.42
E_r	0.40 / 0.65	0.39 / 0.66	0.39 / 0.65	0.23 / 0.44	0.38 / 0.64

Table 5.11 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l’itération 125. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l’IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j} \tau_{ij}$ et des grandeurs scalaires Π et Ψ .

LES construit à partir de l’IA serait viable sur le court voire moyen et long terme. Dans le tableau Tab.(5.11), nous renseignons les scores globaux calculés pour les composantes des grandeurs propres aux niveaux vectoriel $\partial_{x_j} \tau_{ij}$ et scalaire Π et Ψ . Le constat aux niveaux vectoriel et scalaire est similaire que celui établi au niveau tensoriel. Les prédictions dans un régime développé sont bien corrélées avec les données DNS mettant en jeu des scores qui sont du même ordre de grandeur que ceux obtenus au tableau Tab.(5.3)⁸. La comparaison avec les scores concernant ces grandeurs construites à partir modèle de Clark, nous permet de déduire qu’a priori, la prédiction du tenseur τ^{ML} est assez précise et lisse pour que les effets de diffusion de la quantité de mouvement générée par les interactions entre échelles résolues et/ou non résolues soient proches de la diffusion correspondante théorique c’est-à-dire la divergence du tenseur $\tau^{\text{FDNS}} = \overline{U_i U_j} - \overline{U_i} \overline{U_j}$. De même, globalement et a priori, les transferts d’énergie prédits vers les échelles non résolues (la dissipation effective Π) sont cohérents avec la formule théorique ; ainsi en est-il pour la diffusion de l’énergie cinétique par interactions entre échelles résolues et non résolues Ψ . Même si les chutes de précisions sont présents proche des bords du cube comme nous le montrons en annexe à partir des les corrélations de Pearson le long des axes de l’espace, cela ne semble par impacter le score global. Nous pouvons d’ailleurs montrer que ces pics ne sont présents que dans les corrélations de Pearson. En effet, si l’on considère les erreurs relatives le long des axes entre τ_{ij}^{FDNS} et τ_{ij}^{ML} figures Fig.(5.22a), (5.22c) et (5.22e), les pics mentionnés n’apparaissent pas. Ces mêmes figures vont également dans le sens de l’isotropie de la prédiction au sein des cubes. Les différentes erreurs relatives le long des axes n’ont pas le même comportement (notamment selon l’axe y figure Fig.(5.22c)) ; en revanche, les magnitudes sont sensiblement les mêmes et nous retrouvons la tendance du réseau à fournir plus précisément les composantes diagonales que les prédictions sur les composantes extra-diagonales (au niveau absolu). Nous observons également un comportement global identique (pour un axe fixé) entre les erreurs de la prédiction et celles de la modélisation (mesurées par rapport à la FDNS). Les magnitudes ne sont définitivement pas les mêmes, mais les similarités dans les comportements nécessiteraient un travail plus approfondi et pourrait mettre au jour la nécessité de paramètres supplémentaires manquant au modèle de Clark et au CNN dans sa version actuelle.

8. Le renvoi fréquent aux résultats sur d’autres itérations voire simulations cherche à souligner l’idée que la performance du réseau à convolutions est robuste aux changements topographiques et en magnitude des champs de vitesse.

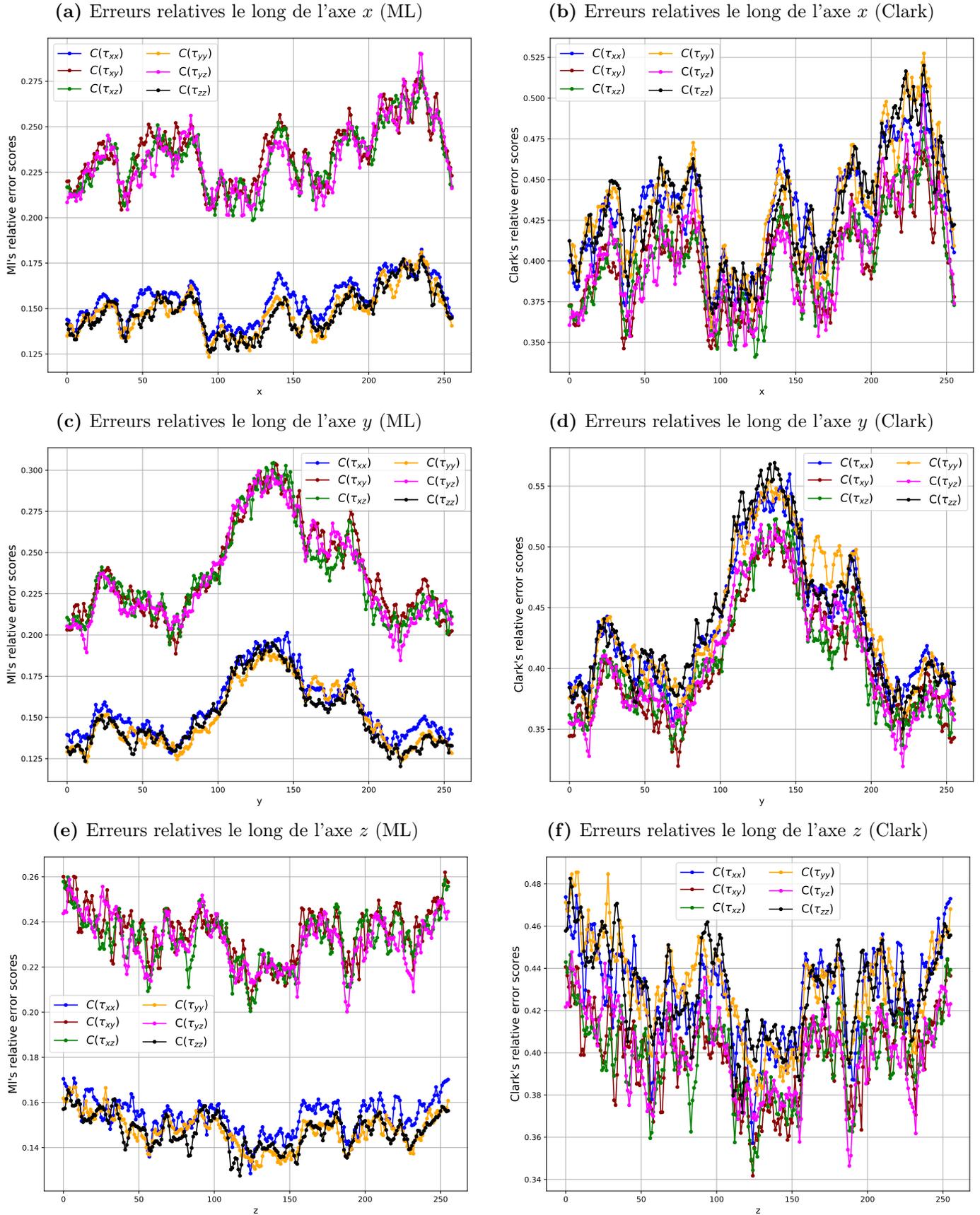


Figure 5.22 – À l'itération 125, erreurs relatives entre τ_{ij}^{FDNS} et τ_{ij}^{ML} (figures de gauche). Idem entre τ_{ij}^{FDNS} et τ_{ij}^{Clark} (à droite).

Nous complétons notre analyse en considérant les PDFs de deux composantes du niveau vectoriel (choisies arbitrairement) et des deux grandeurs associées au niveau scalaire. Comme pour les figures Figs.(5.21), nous traçons les composantes issues des tenseurs τ^{FDNS} , τ^{ML} et τ^{Clark} . Les deux figures du haut montrent les tracés des composantes x et z du vecteur $\partial_{x_j}\tau_{ij}$ et l'on remarque que les distributions des grandeurs issues de la modélisation et celles de la prédiction sont proches, avec une meilleure précision sur l'ensemble des événements pour les grandeurs issues de la prédiction. Les comparaisons sur les PDFs des grandeurs scalaires Π et Ψ sont tracées dans les figures du bas Figs.(5.23c) et (5.23d). Dans ces dernières, on remarque que la modélisation de Clark surestime la probabilité d'obtenir des événements les plus communs notamment pour Π . Concernant la grandeur Ψ on peut voir que ces événements sont sur-représentés dans les distributions issues du modèle de Clark et de la prédiction. L'intermittence est globalement mieux représentée par les grandeurs issues de la prédiction sous entendant que les événements les plus énergétiques sont dans les gammes de valeurs attendues ; ce qui n'est pas toujours le cas pour les grandeurs vectorielles et scalaires issus de τ_{ij}^{Clark} . Les coupes des champs physiques nous ont semblé superflues à cette itération puisque globalement les résultats sont très proches de ceux obtenus précédemment. Les grandeurs explicitent beaucoup plus de

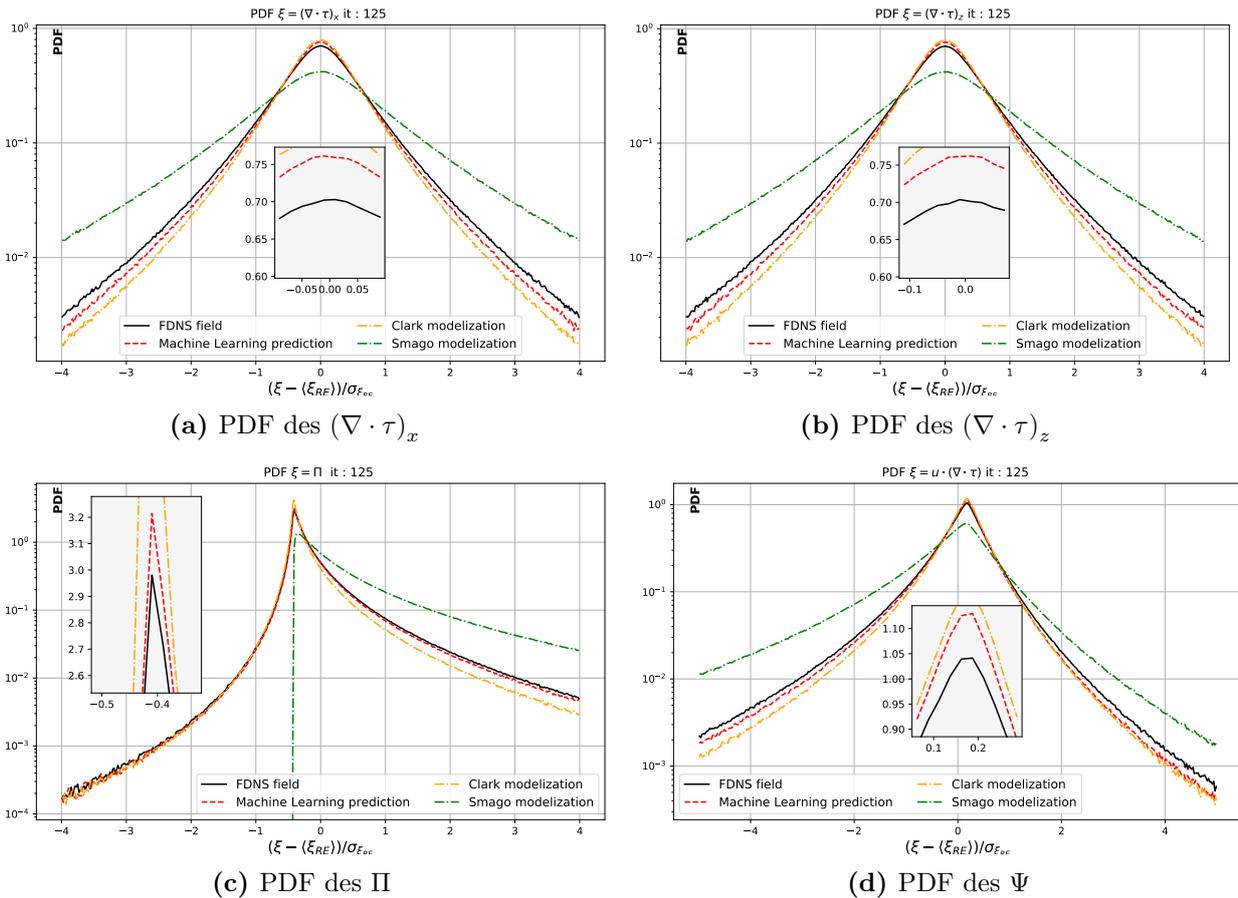


Figure 5.23 – À l'itération 125, comparaisons des PDFs des composantes x et z de $\partial_{x_j}\tau_{ij}$ ainsi que des scalaires Π et Ψ théoriques (courbes continues noires), prédites et issues des modèles Clark et de Smagorinsky (ligne brisée rouge, lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS.

structures d'amplitudes bien différentes. L'analyse des termes de la décomposition de Leonard n'est également pas menée pour cette itération, néanmoins nous fournissons dans les figures en fin de section les corrélations de Pearson des 6 composantes de L_{ij}^{mod} , C_{ij}^{mod} et R_{ij}^{mod} où mod représente les modèles augmenté ou algébrique.

En conclusion, nous avons effectué l'analyse des performances de l'IA dans un régime turbulent de Reynolds Re_λ près de 200% plus élevé que dans les itérations considérées lors de l'apprentissage, et malgré tout les performances relevées sont quasi identiques à celles obtenues lors des tests précédents et ceux malgré des champs radicalement différents voir figures Fig.(5.20). Nous avons donc mis en évidence à deux reprises que même pour des champs de vitesse plus riches en structures, en intrications de ces dernières et de résolutions différentes, le réseau a pu fournir des instantanées du tenseur τ_{ij}^{ML} qualitativement et quantitativement très satisfaisantes. Enfin, le défi de surpasser le modèle de Clark fixé depuis la fin du paragraphe 2.3.2 est atteint pour la gamme de régime pour lequel l'énergie cinétique et la dissipation sont proches de leur valeur asymptotique, englobant le régime développé. Nous pouvons conclure que l'IA a découvert une relation générale entre le champ de vitesse filtrée et les contraintes attendues par les structures qui le composent. Ce constat est vrai pour une prédiction instantanée.

Dans les sections suivantes, nous voulons confirmer que cette relation est également vraie au delà du régime développé de la turbulence, en rajoutant une nuance *physique* entre les données utilisées dans l'apprentissage et les données fournies dans ces sections. Nous remontons la courbe de l'évolution cinétique et de la dissipation figure (5.18). Au plus nous remontons ces deux courbes au plus nous entrons dans des régimes turbulents différents. Le point à l'itération 88 se place dans le régime où les petites échelles dissipent l'énergie des grandes échelles plus rapidement qu'elles ne sont produites.

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.98 / 0.91	0.97 / 0.91	0.97 / 0.91	0.98 / 0.91	0.97 / 0.91	0.98 / 0.91
$1 - R^2$	0.04 / 0.30	0.06 / 0.18	0.06 / 0.18	0.04 / 0.31	0.06 / 0.18	0.04 / 0.31
E_r	0.17 / 0.45	0.25 / 0.43	0.25 / 0.43	0.16 / 0.45	0.25 / 0.43	0.17 / 0.45

Table 5.12 – Scores obtenus sur la prédiction au niveau tensoriel à l'itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes τ_{ij} .

5.2.2 Prédiction dans le régime dissipatif : étude des performances et des transferts d'énergie à travers la décomposition de Leonard

Comme précédemment nous présentons les performances de la prédiction (niveau tensoriel) à partir des trois scores ; les scores sur τ_{ij}^{Clark} sont également renseignés en orange dans le tableau Tab.(5.12). Si les performances du ML sont quasi similaires à celles présentées dans le régime développé Tab.(5.10), ce n'est pas le cas pour la modélisation de Clark : près de 50% de la variance est perdue sur les composantes diagonales ; alors que seulement 6% de la variance n'est pas recouverte par la prédiction. Sur les composantes extra-diagonales près 20% de la variance n'apparaît pas dans les distributions des τ_{ij}^{Clark} ; 6% n'est pas recouverte par la prédiction. Des constatations similaires peuvent être faites à propos des erreurs relatives et Pearson commises par la modélisation et l'IA par rapport aux données théoriques. Nous confirmons ces résultats le long d'un des trois axes⁹. Ces tracés sont présentés figures Figs.(5.24). L'écart relatif aux composantes FDNS permet de rendre compte de la qualité des prédictions en tant que telle, mais également par rapport à la modélisation de Clark.

Aux niveaux vectoriel et scalaire, nous renseignons les scores globaux Tab.(5.13). Ici aussi, les

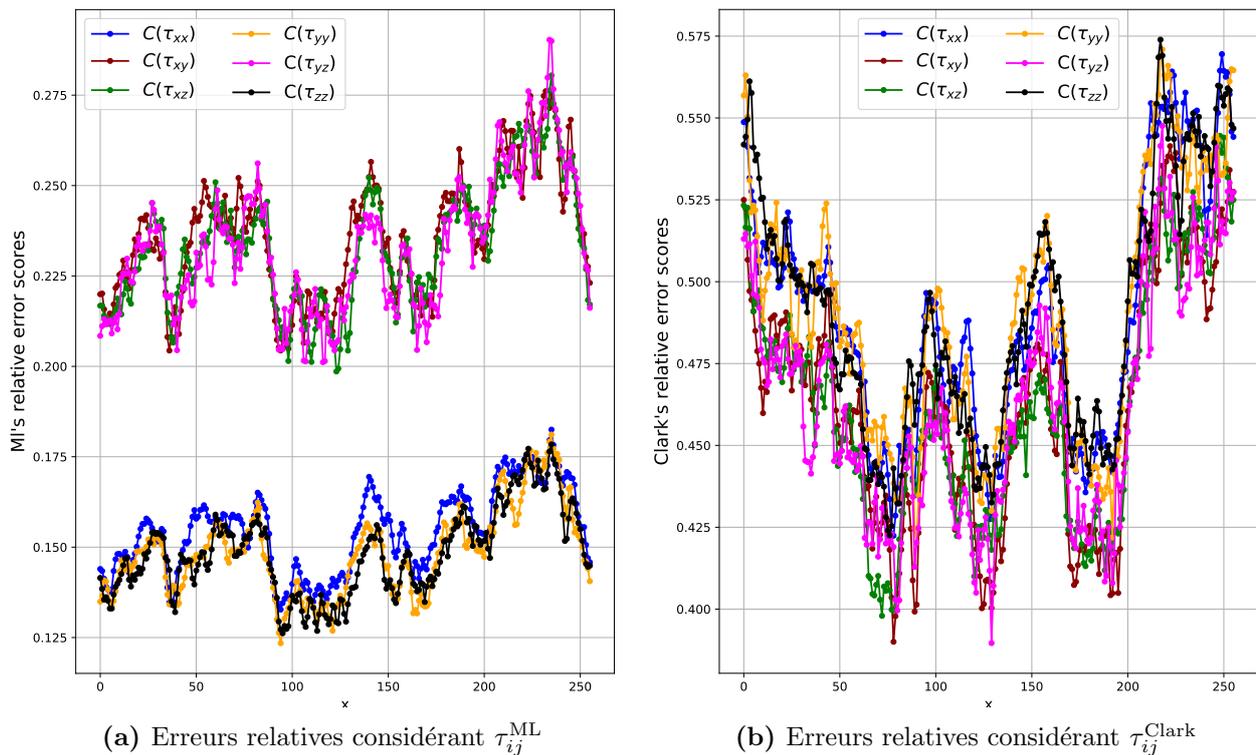


Figure 5.24 – Erreurs relatives le long de l'axe x . Les courbes bleue, noire et orange représentent les écarts sur les composantes diagonales ; les courbes rouge foncé, verte et magenta représentent les écarts sur les composantes extra-diagonales. Les écarts sont calculés à l'itération 88.

9. Nous ne considérons qu'un seul des trois axes entendu l'isotropie de la prédiction mise en évidence dans les paragraphes précédents.

Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.91 / 0.71	0.91 / 0.71	0.91 / 0.71	0.96 / 0.88	0.91 / 0.72
$1 - R^2$	0.17 / 0.50	0.17 / 0.50	0.17 / 0.50	0.07 / 0.28	0.17 / 0.49
E_r	0.41 / 0.71	0.41 / 0.71	0.41 / 0.71	0.25 / 0.49	0.40 / 0.69

Table 5.13 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l’itération 88. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l’IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j} \tau_{ij}$ et des grandeurs scalaires Π et Ψ .

valeurs de ces métriques sur les grandeurs construites à partir de τ_{ij}^{ML} sont quasiment identiques à celles obtenues à l’itération 125. Ainsi les corrélations de Pearson sont toutes supérieures à 90% et seulement un sixième de la variance n’est pas recouverte par la prédiction. Concernant les scores sur les grandeurs issues de la modélisation de Clark, les erreurs relatives sont proches des 75 % sur chacune des composantes du vecteur $\nabla \cdot \tau$ s’en suit des corrélations de Pearson de l’ordre de 70% et une perte de la moitié de la variance. Pour le scalaire Π c’est quatre fois plus de variance perdue par le modèle par rapport à la prédiction près de trois plus pour le scalaire Ψ . Pour éviter toute redondance, nous proposons de tracer les PDFs des grandeurs suivantes τ_{xx} , τ_{yz} , $(\nabla \cdot \tau)_y$, $(\nabla \cdot \tau)_z$, Π et Ψ , couvrant les composantes diagonales/extra-diagonales des tenseurs, deux composantes du niveau vectoriel et les deux champs scalaires relatifs à l’évolution de l’énergie cinétique au sein de l’écoulement. Ces tracés sont effectués figures Figs.(5.25).

L’interprétation des différentes distributions est *globalement* la même que précédemment. Les distributions issus des grandeurs ML retrouvent les événements rares (l’intermittence) comme les événements les plus courants, et ce pour quasiment toutes les grandeurs considérées, englobant les autres composantes tensorielles et vectorielle. Les distributions retrouvées par les grandeurs issues de la modélisation de Clark sont presque systématiquement bien différentes des distributions attendues notamment au niveau tensoriel et scalaire.

En étant plus critique à l’égard des grandeurs prédites et issues de ces dernières, nous remarquons plusieurs décrochages inédits notamment pour les composantes diagonales du tenseur τ_{ij}^{ML} . Aux autres niveaux, on en remarque de plus prononcées que précédemment, signe que les occurrences des signaux rares et énergétiques dans les grandeurs prédites et construites à partir de ces dernières sont moindres.

Étude des composantes de Leonard - Nous complétons notre analyse avec l’étude des composantes de la décomposition de Leonard, pour déterminer si pour cette simulation encore, l’IA permet de modéliser des interactions entre les échelles non résolues dont le modèle Clark est complètement aveugle. Nous rappelons que nous considérons la décomposition (2.3.6) qui est exacte pour la DNS et la théorie. Dans notre cas, nous décomposons τ_{ij}^{mod} en trois composantes L_{ij}^{mod} , C_{ij}^{mod} et R_{ij}^{mod} qui sont construites indépendamment les unes des autres selon les formules

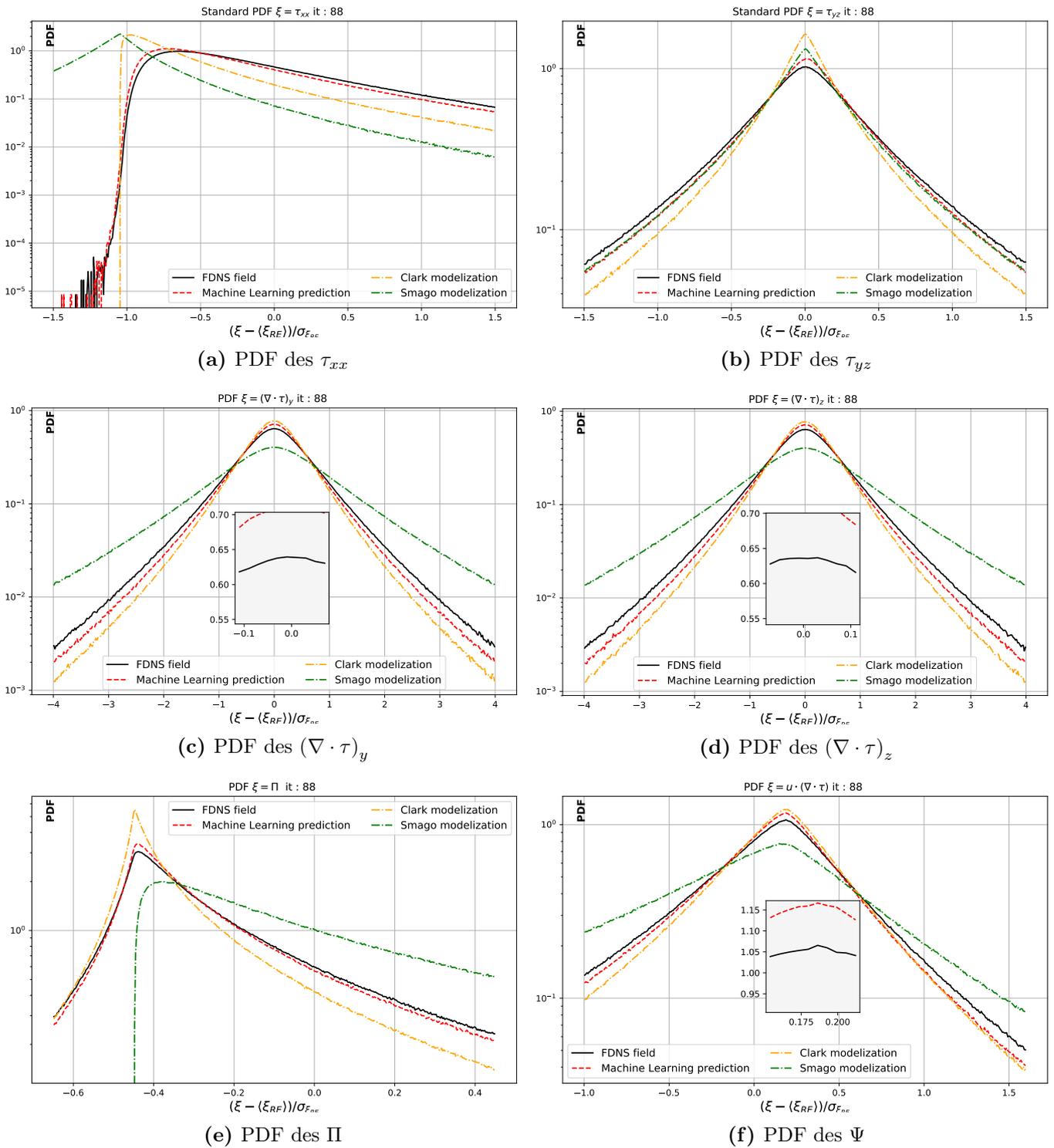


Figure 5.25 – À l'itération 88, comparaisons des PDFs des composantes τ_{xx} et τ_{yz} , $(\nabla \cdot \tau)_y$ et $(\nabla \cdot \tau)_z$ ainsi que des scalaires Π et Ψ calculées théoriquement (courbe solide noire), prédites (ligne brisée rouge) et modélisées (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS.

Eqs.(2.4.1), (2.4.2) et (2.4.3) respectivement telles que

$$\tau_{ij}^{\text{Mod}} \neq L_{ij}^{\text{Mod}} + C_{ij}^{\text{Mod}} + R_{ij}^{\text{Mod}}$$

Scores	R_{xx}	R_{xy}	R_{xz}	R_{yy}	R_{yz}	R_{zz}
Pearson	0.74 / -0.64	0.62 / -0.52	0.62 / -0.52	0.75 / -0.64	0.62 / -0.52	0.76 / -0.64
$1 - \mathcal{R}^2$	0.61 / > 1	0.73 / > 1	0.71 /> 1	0.54 / > 1	0.72 / > 1	0.54 / > 1
E_r	0.51 / 1.43	0.85 / 1.49	0.85 / 1.49	0.5 / 1.43	0.85 / 1.49	0.5 / 1.43

Table 5.14 – Scores obtenus sur les composantes R_{ij} calculées selon la formule Eq.(2.4.3), pour l’itération 88. Les valeurs en noir : scores sur R_{ij}^{ML} ; en orange scores sur R_{ij}^{Clark} .

L’idée derrière l’étude de ces grandeurs est en fait de mesurer à quel point chaque composante s’éloigne des composantes de références. Les composantes L_{ij} et C_{ij} construites à partir de τ_{ij}^{Clark} ou de τ_{ij}^{ML} sont très corrélées avec leur équivalent issus des formules respectives (2.3.7) et (2.3.8), nous renseignons les tableaux des scores sur les 6 composantes dans l’annexe Tabs.(B.3) et (B.4). Comme précédemment la composante R_{ij} renferme plus d’informations, plus de d’intrigue. Nous rapportons les scores globaux des corrélations entre R_{ij}^{ML} vs R_{ij}^{FDNS} (c’est-à-dire R_{ij}) et R_{ij}^{Clark} vs R_{ij}^{FDNS} . Tout d’abord on remarque que le tenseur $\mathbf{R}^{\text{Clark}}$ construit à partir de la modélisation de Clark est fortement décorrélé d’avec le tenseur théorique. Par contre, le même tenseur construit à partir de τ_{ij}^{ML} présente des corrélations encore plus fortes que celles obtenues précédemment section 5.1.2. Cette fois, les composantes extra-diagonales sont bien mieux retrouvées au point où la perte de variance passe en dessous des 100 % pour se situer autour des 75%, idem pour l’erreur relative mesurée pour ces mêmes composantes qui plafonne à 85% contre 110%. Même si ces valeurs sont grandes, elles sont malgré tout encourageantes et exigent d’en approfondir leur analyse en traçant les PDFs d’une composante diagonale R_{yy} et et d’une extra-diagonale R_{xz} pour rester concis et cohérent avec la régularité des scores respectivement aux types de composantes. Ces tracés sont effectués figures Figs.(5.26). Elles ressemblent beaucoup aux courbes figures Figs.(5.11) à ceci près que les distributions sur les composantes extra-diagonales du tenseur \mathbf{R}^{ML} sont très proches de celles théoriques avec un très léger décrochement au niveau des intermittences alors que ces mêmes composantes du tenseur $\mathbf{R}^{\text{Clark}}$ sont moins proches que précédemment. De manière générale, la distribution des composantes R_{ii}^{ML} sont moins performantes que celles obtenues section 5.1.2 ; les corrélations sont également moins élevées. Nous proposons des coupes des champs physiques de normale $x = 3$ de ces deux mêmes composantes pour chacun les champs théoriques, prédits et modélisés au sens de Clark. Pour mieux apprécier les différences entre modélisations et prédictions, nous avons changé la carte des couleurs selon les deux composantes. Commençons par analyser les coupes de la composante diagonale figures Figs.(5.27). Nous retrouvons l’incapacité de modéliser les composantes diagonales du tenseur des contraintes de la part du modèle de Clark. Plus que ça, il s’avère que les valeurs modélisées sont négatives conformément à la distribution obtenue en orange sur la figure Fig.(5.26a) dans laquelle la quasi totalité des événements étaient situés dans l’hémisphère gauche du graphique. Sans la visualisation Figs.(5.27) il était impossible de conclure à cause du centrage de la PDF. A contrario, les composantes τ_{ij}^{ML} sont topographiquement semblables à celles attendues. De plus, on peut voir figure du bas au centre, que les

erreurs fortes sont assez rares si bien que presque la totalité des erreurs sont entre 10 et 25% ; ce qui était attendu au vu des corrélations de Pearson. Également, la majorité des événements de R_{yy}^{ML} exhibés Fig.(5.27) sont proches et dans l'hémisphère droit, le décrochage aux niveaux des queues positives peut se traduire physiquement par les fortes erreurs localisées.

Nous pouvons mener ce même genre d'analyse sur les coupes des composantes extra-diagonales théoriques, prédites et modélisées ainsi que les erreurs associées figure Figs.(5.26b). Les faibles différences d'amplitude dans cette composante laissent supposer que le modèle fait mieux pour ce type de composantes. Or il a été montré et nous le confirmons ici que ce n'est pas le cas. La tendance à sous évaluer les composantes de \mathbf{R} est cachée par le fait que les valeurs extra-diagonales sont faibles. Ceci dit, nous remarquons malgré tout que la topographie fait apparaître certaines zones d'activité ressemblant globalement à celle des champs attendus, aux magnitudes près donc. L'écart en valeurs absolues représenté figure du bas à droite met en exergue que les erreurs au dessus de 100% sont courantes voire parfois 150% aux zones les plus actives du champ théorique. Le champ construit à partir de τ_{ij}^{ML} exhibe non seulement une topographie très similaire à celle de R_{xz}^{FDNS} mais semble également bien rendre compte des amplitudes de contraintes aux points les plus actifs. Les erreurs associées figure du bas au centre montrent de larges écarts de façon très localisée, si bien que la majorité des erreurs se situent autour de 30%, avec toutefois certaines zones (hors pics) se rapprochant des 75% d'erreur. Cette analyse est cohérente avec les figures précédentes et notamment la figure Fig.(5.26b) dans laquelle les événements les plus énergétiques sont sous légèrement représentés dans la distribution de R_{xz}^{ML} et globalement des deux autres composantes extra-diagonales. L'erreur plus diffuse retrouvée dans la coupe de R_{xz}^{Clark} est également prévisible puisque les événements quasi-nuls sont sur-représentés contrairement aux événements les plus énergétiques qui sont très rares et de toute façon loin des valeurs attendues.

Comme précédemment nous pouvons considérer les écarts des termes dissipatifs associés à aux termes de la décomposition de Leonard pour les modèles augmenté et algébrique par rapport aux termes dissipatifs attendus Π_{Lij} , Π_{Cij} et Π_{Rij} . Rappelons simplement que pour la prédiction

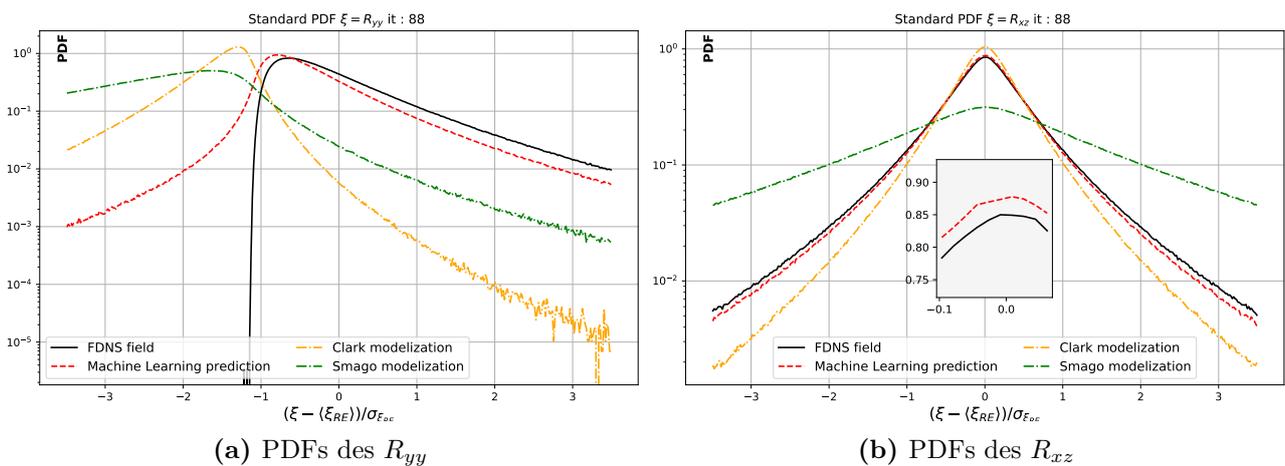


Figure 5.26 – À l'itération 88, comparaisons des PDFs de R_{yy} et R_{xz} construites à partir de τ_{ij}^{FDNS} (courbe continue noire), de τ_{ij}^{ML} , de τ_{ij}^{Clark} et de τ_{ij}^{Smago} (courbes pointillées resp. rouges, oranges et vertes). Les PDFs sont standardisées par les moments de R_{ij}^{FDNS} .

ou la modélisation, la somme de ces trois termes dissipatifs n'est pas égale au tenseur global Π^{mod} par construction. Nous renseignons les scores globaux obtenus par rapport aux grandeurs théoriques dans le tableau Tab.(5.15). Ces scores sont très proches de ceux présentés Tab.(5.7) pour des comparaisons similaires. On retrouve alors que les dissipations associées aux termes L_{ij}^{mod} et C_{ij}^{mod} ne diffèrent que de très peu des quantités équivalentes théoriques. Les scores pour la prédiction sont quasiment identiques ; la modélisation de Clark est cependant légèrement moins précise que précédemment, notamment au niveau de la variance qui manque deux fois plus d'informations et de l'erreur relative plus de 50% plus élevée pour ces dissipations. Les résultats sur la dissipation due aux interactions purement sous-maîles sont différents. D'une part la prédiction ne perd plus que 66% de la variance contre plus de 100% précédemment, et l'erreur relative globale est évaluée à 75% contre 110%. D'autre part, il semble que les grandeurs construites utilisant τ_{ij}^{Clark} n'ont pas gagné en précision. La variance est toujours totalement perdue, la corrélation de Pearson est toujours négative et les erreurs relatives plafonne à 150%.

Nous proposons de comparer les PDFs obtenues à partir des différents $\Pi_{\mathbf{R}}$ figure Fig.(5.29). Remarquons tout d'abord, que le modèle de Smagorinsky présente majoritairement des zones de sur-dissipation par rapport au comportement attendu. En visualisant des coupes du champ $\Pi_{\mathbf{R}}^{\text{Smago}}$ nous nous sommes aperçus que les erreurs dépassaient les 900% (expliquant pourquoi nous n'en avons pas beaucoup parlé dans cette partie). La distribution obtenue à partir de ce que nous avons noté $\Pi_{\mathbf{R}}^{\text{Clark}}$ est majoritairement différente de la cible. L'intermittence comme

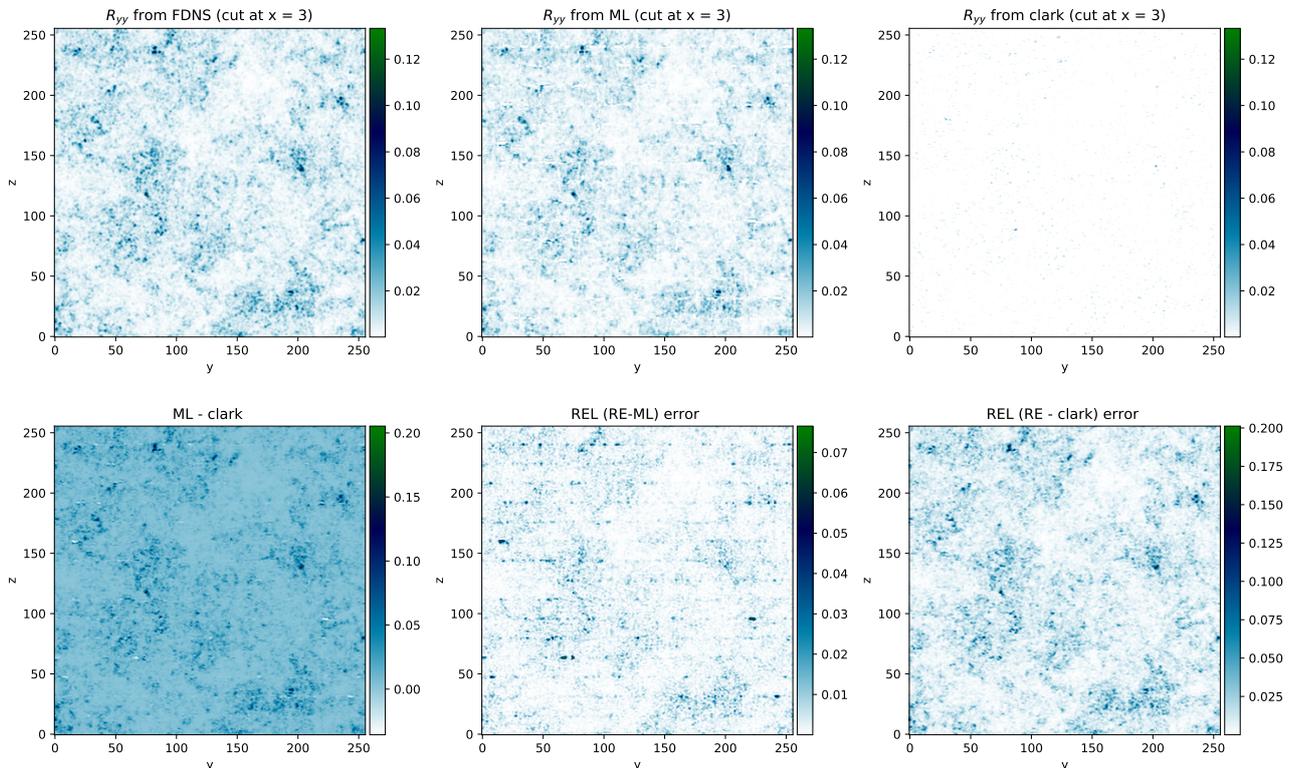


Figure 5.27 – **Figures du haut** : Coupe de la composante R_{yy} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$.

Scores	Π_{Lij}	Π_{Cij}	Π_{Rij}
Pearson	0.99 / 0.97	0.99 / 0.96	0.66 / -0.58
$1 - \mathcal{R}^2$	0.02 / 0.08	0.02 / 0.08	0.66 / > 1
E_r	0.14 / 0.28	0.14 / 0.28	0.76 / 1.5

Table 5.15 – Scores obtenus sur les dissipations reliées aux termes de la décomposition de Leonard pour l’itération 88. En noir et en orange, les scores évalués sur les grandeurs issues de τ_{ij}^{ML} et celles issues de τ_{ij}^{Clark} respectivement.

les événements plus courants ne se retrouvent pas du tout. Au niveau de la distribution des valeurs composant $\Pi_{\mathbf{R}}^{ML}$, nous constatons que la fréquence des événements les plus courants est bien retrouvée ; aux niveaux des événements les plus énergétiques le constat est plus mitigé, mais globalement meilleur que le modèle de Clark.

Analyse de la rétrodiffusion des modèles - Pour aller plus loin nous proposons une analyse inédite par rapport à celle établie section 5.1.2. Rappelons que l’interprétation de $\Pi > 0$ correspond au transfert de l’énergie cinétique depuis les échelles résolues vers les échelles non

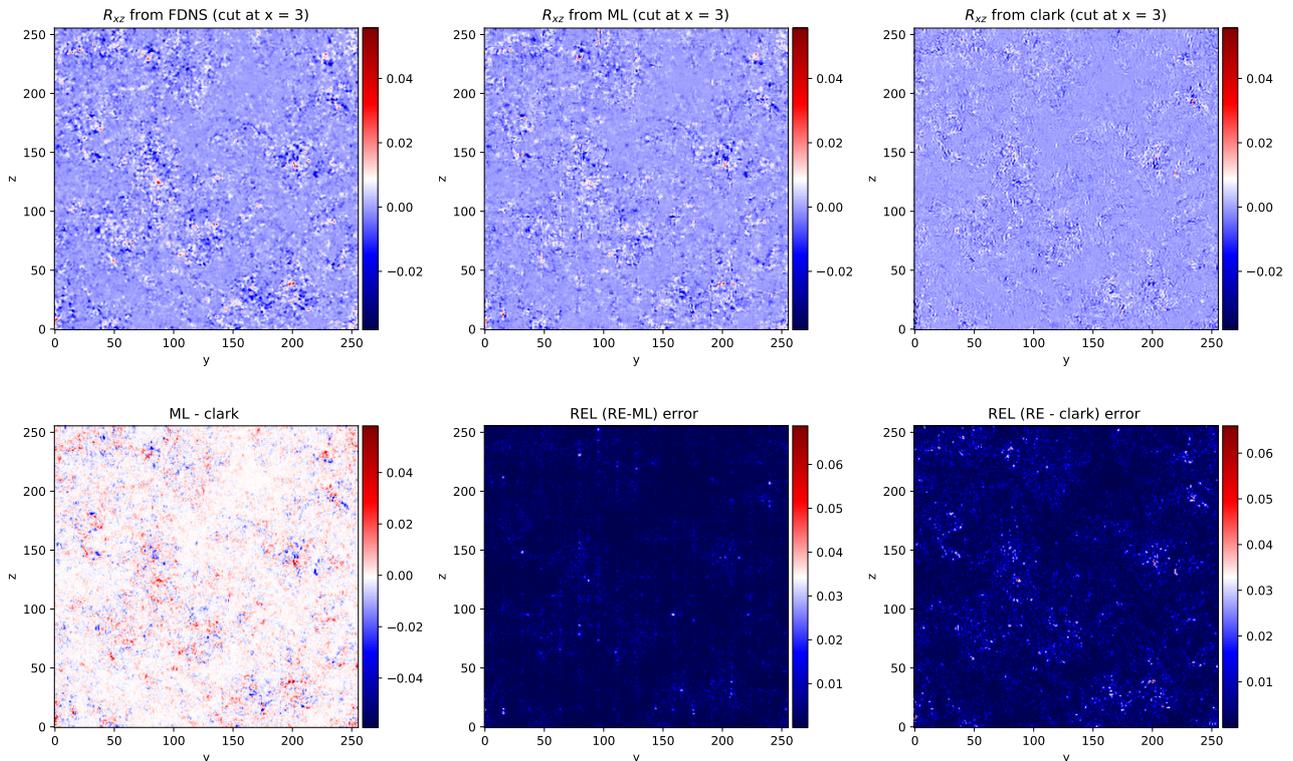


Figure 5.28 – **Figures du haut** : Coupe de la composante R_{xz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). **Figures du bas** : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $x = 3$.

résolues ; c'est ce que l'on a appelé la dissipation effective puisque conformément à l'approche LES, une partie de l'énergie est dissipée à la coupure (du point de vue des échelles résolues). Sans le détailler outre mesure, nous avons également mentionné que les rares cas pour lesquels cette dissipation était négative correspondaient à un transfert d'énergie dans la direction inverse, aussi appelé phénomène de rétro-diffusion. Nous utilisons la définition de ce phénomène pour l'étudier de façon intuitive. Sur chacun des points de discrétisation du domaine, nous comptabilisons les valeurs positives et négatives de ce scalaire. Nous ramenons ces résultats sur le nombre total de points de discrétisation pour obtenir un pourcentage des effets dont Π sera responsable. Le challenge est ensuite de comparer les dissipations effectives relatives aux types d'interactions décrites par $\Pi_{L_{ij}}$, $\Pi_{C_{ij}}$ et $\Pi_{R_{ij}}$ **indépendamment** les unes des autres ne prenant donc pas en compte les éventuels télescopages. Nous renseignons les résultats dans le tableau Tab.(5.16).

Concrètement, ce tableau nous indique qu'il y a le même pourcentage de valeurs négatives dans les scalaires Π et Π^{ML} et que cette part est sur-évaluée par la modélisation de Clark. En couplant cette observation avec les corrélations ainsi que les distributions Figs.(5.25e), nous assurons

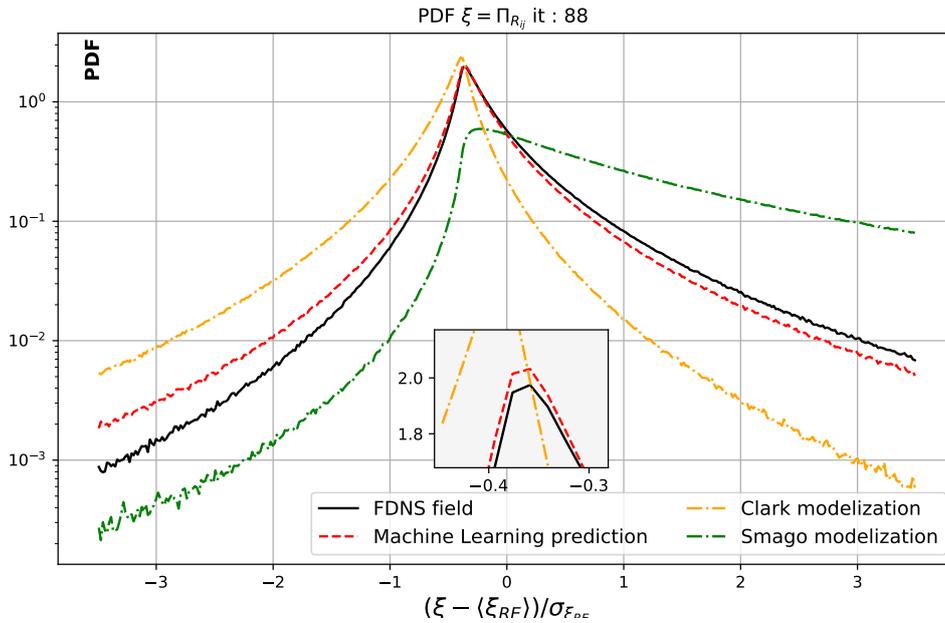


Figure 5.29 – À l'itération 88, comparaisons des PDFs des champs scalaires $\Pi_{\mathbf{R}}^{\text{FDNS}}$ (courbe continue noire), avec $\Pi_{\mathbf{R}}^{\text{ML}}$ (ligne brisée rouge), $\Pi_{\mathbf{R}}^{\text{Clark}}$ et $\Pi_{\mathbf{R}}^{\text{Smago}}$ (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS.

ξ	Card ($\{\xi < 0\}$)
Π	20% / 20% / 27%
Π_L	46% / 48% / 51%
Π_C	45% / 46% / 50%
Π_R	26% / 31% / 63%

Table 5.16 – Pourcentages de rétrodiffusion pour Π calculé selon la formule théorique et celles relatives à chacune des composantes de la décomposition de Leonard. En noir les valeurs théoriques, en rouge et orange celles construites à partir de τ_{ij}^{ML} et τ_{ij}^{Clark} respectivement.

que les zones de rétrodiffusion présentent dans Π et Π^{ML} sont situées dans les mêmes régions (corrélations de Pearson proche des 95%) mais son également de magnitudes similaires (léger décrochage aux niveaux des queues notamment celles de droite de la figure sus-mentionnées). Ces mêmes tableaux et figures montrent qu’au niveau de la répartition spatiale, des magnitudes et de la fréquence d’occurrence de la rétro-diffusion, le modèle de Clark n’est pas aussi voire pas suffisant précis. Ce constat est connu de la littérature (Balarac et al., 2013; Leonard, 1975) et cette sur-estimation de la rétrodiffusion déstabilise le modèle sur le court-moyen terme. Les trois dernières lignes du tableau Tab.(5.16) montrent les pourcentages de valeurs négatives au sein des grandeurs Π_L , Π_C et Π_R calculées indépendamment les unes des autres. Ce tableau confirme un résultat que nous avons émis concernant l’origine des erreurs de modélisation du modèle de Clark : c’est la composante relative aux intersections purement sous-maille qui estimant de trop la rétrodiffusion et semble baisser la performance générale du modèle du point de vue énergétique.

À l’itération précédente, nous avons montré que dans le régime développé pour lequel le champ de vitesse exprime le plus de complexité, le CNN réussit à prédire corrélés à plus de 97% avec les tenseurs théoriques assez précis pour permettre de construire des vecteurs et des scalaires très précis, surpassant le modèle du gradient sur toutes ces considérations. Cette itération ajoute une nuance supplémentaire et permet d’étendre le scope des écoulements turbulents à partir desquels le réseau peut prédire le tenseur des interactions sous-maillages. L’écoulement encore dans son régime transitoire et sur-dissipatif est foncièrement différent aussi bien au niveau des mécanismes à l’œuvre que dans la topographie qu’il arbore. Malgré tout, le réseau s’est montré robuste et consistant sur presque tous les aspects. Cette état transitoire a été également un candidat intéressant pour l’étude des dissipations propres aux différentes interactions qui composent τ_{ij} . Il a été montré que la précision de la prédiction permettait de prendre en compte une partie de la Physique à laquelle tout modèle LES “classique” est aveugle par construction. Ainsi le fait d’apprendre à décortiquer le champ de vitesse filtré afin de prédire les contraintes sous-maillages permet de constituer une image plus globale de cette Physique justifiant en soi l’utilisation des intelligences artificielles dans l’élaboration des modèles LES.

La dernière itération que nous considérons avec détails est celle correspondant au régime dans lequel les échelles sont encore créées et pour lequel les effets dissipatifs sont plus faibles ou du moins sous-dominants par les effets productions. Les champs de vitesse font donc apparaître des structures plus grandes. Ce qui implique que chaque cube de discrétisation fourni au réseau peut contenir moins d’information sur l’état de la turbulence. Évaluons alors notre modèle sur ce cas inédit.

5.2.3 Prédiction dans le régime productif : étude des performances et des transferts d’énergie à travers la décomposition de Leonard

Comme pour toutes les itérations que nous avons considérées ici nous commençons par l’analyse des scores globaux retranscrits dans le tableau Tab.(5.17). On retrouve les excellents scores de la prédiction avec près de 0.98% de corrélations avec les composantes cibles τ_{ij}^{FDNS} , et seule-

Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.98 / 0.91	0.97 / 0.91	0.97 / 0.91	0.98 / 0.91	0.97 / 0.91	0.98 / 0.91
$1 - \mathcal{R}^2$	0.04 / 0.30	0.06 / 0.18	0.06 / 0.18	0.04 / 0.31	0.06 / 0.18	0.04 / 0.31
E_r	0.17 / 0.45	0.25 / 0.43	0.25 / 0.43	0.16 / 0.45	0.25 / 0.43	0.17 / 0.45

Table 5.17 – Scores obtenus sur la prédiction au niveau tensoriel à l’itération 61. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l’IA et par le modèle Clark sur les composantes τ_{ij} .

ment 5% de la variance perdue en moyenne sur les six différentes composantes. Les erreurs relatives de l’ordre de 15% et de 25% sur les composantes diagonales et extra-diagonales sont retrouvées à cette itération également. Ces erreurs sont relatives aux gammes d’amplitudes que les composantes expriment. De façon consistante avec les itérations précédentes étudiées pour cette simulation, les modélisations du modèle de Clark sont moins précises et en particulier par rapport à la variance perdue (jusqu’à 31% pour les composantes diagonales) et des erreurs relatives 2 à 3 fois plus fortes que celles calculées en considérant τ_{ij}^{ML} . On peut donc d’ores et déjà remarquer la robustesse de la prédiction par rapport aux entrées de nature, de niveau de détail et de topographie différentes. Ceci de façon absolue mais également relative au modèle de Clark dont les performances sont en deça du modèle augmenté par les données.

Nous évaluons les performances de ces deux modèles aux niveaux vectoriel et scalaire pour évaluer à ce moment de l’écoulement l’impact des modélisations respectives, sur l’évolution du champ de vitesse et de l’énergie cinétique. Nous évaluons donc les scores globaux sur les composantes du vecteur $\nabla \cdot \boldsymbol{\tau}$ et des scalaires Π et Ψ et renseignons ces valeurs dans le tableau Tab.(5.18). Nous constatons également que les scores sont du même ordre de grandeur que précédemment en ce qui concerne les grandeurs issues des prédictions. Le modèle du gradient est être moins précis que précédemment avec des scores à l’ordre un et deux significativement moins bons que ceux de l’IA.

Fidèle à nos investigations passées, nous traçons figures Figs.(5.30) les PDFs des modélisations augmentées et algébriques des composantes τ_{zz} et τ_{xy} (niveau tensoriel), des composantes selon \vec{x} et \vec{y} de la grandeur vectorielle puis des scalaires Π et Ψ . Ici encore bien que les champs soient complètement différents (voir notamment les figures (5.20c) et (5.20d)) les distributions des prédictions sont quasiment totalement identiques aux distributions théoriques et ceci couplé avec les scores précédents permet d’établir que la prédiction du tenseur $\boldsymbol{\tau}^{\text{ML}}$ assure une modélisation consistante aux différents niveaux indiqués par (Ferziger and Reynolds, 1979).

Conclusion de cette section et du chapitre - Cette section constituait le test le plus abouti que nous avons considéré. D’une part parce qu’il mettait en jeu des champs de vitesse fondamentalement plus riches et plus complexes que ceux avec lesquels l’IA a été entraînée. Depuis l’établissement des bases de données section 4.2.1 la question de savoir si une IA entraînée

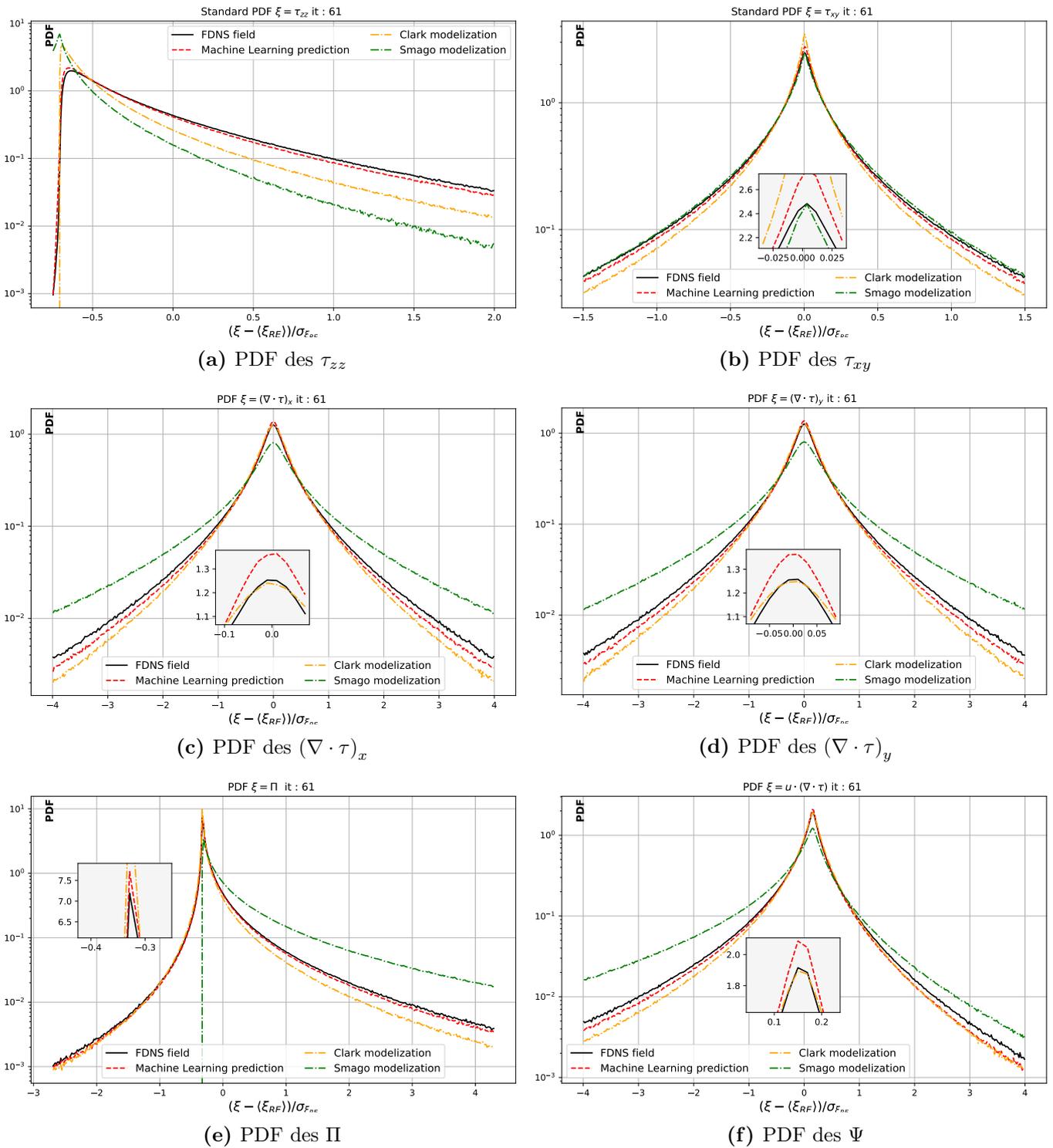


Figure 5.30 – À l’itération 61, comparaisons des PDFs des composantes τ_{zz} et τ_{xy} , $(\nabla \cdot \tau)_x$ et $(\nabla \cdot \tau)_y$ ainsi que des scalaires Π et Ψ calculées théoriquement (courbe solide noire), prédites (ligne brisée rouge) et modélisées (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moments de la FDNS.

avec si peu de données allait être capable de généraliser ses excellents scores sur l’ensemble

d'apprentissage et de test au delà du cadre qui a servi à l'optimisation des poids. La réussite et la constance des performances du CNN révèlent rétroactivement que l'apprentissage était non seulement suffisant mais assez efficace pour découvrir ce qu'il y a de plus universel entre le champ de vitesse filtré et les contraintes générées par la présence de coupure, nécessaire pour fermer les équations de Navier-Stokes. L'expérience acquise dans les tentatives pour établir un modèle efficace, nous laisse penser que la discrétisation en cubes a fortement contribué à la découverte de cette quintessence au sein des données, et ce en des temps relativement court, aspect qui sera d'ailleurs abordé légèrement plus tard dans cette discussion. Un autre point fondamental tout au long des travaux présentés est la conservation du facteur $R = 2$ qui constitue en même temps la force de notre approche et sa faiblesse, en tout cas pour les modèles LES tels que nous les connaissons. D'autre part, les itérations considérées dans cette section étaient plus proches des zones transitoires que les modèles ont vraisemblablement plus de mal à décrire. En effet, même si l'itération 42 de la simulation précédente (sujet de la section 5.1.3) se situait exactement au pic d'énergie cinétique, il était bien loin du pic de dissipation qui marque véritablement l'entrée en action des échelles dissipatives. Ces résultats étaient en revanche nécessaires pour ouvrir la voie et émettre des conjectures sur les capacités de généralisation du CNN, sa robustesse et sa constance. Nous nous sommes donc fixé comme objectif de pousser encore plus loin l'analyse des capacités de l'IA en se rapprochant des zones généralement instables et délicates pour la modélisation. Le succès constaté sections 5.1.2 et 5.2.3 permet d'étendre toutes les conclusions et conjectures intermédiaires.

Cette analyse a priori, ne s'est pas limitée aux niveaux tensoriel, vectoriel et scalaire préconisés par la littérature ; nous avons voulu apporter un premier élément concernant l'interprétation des meilleures performances de la prédiction par rapport à la modélisation de Clark, à commencer par le niveau tensoriel. Pour cela, nous avons isolé puis analysé chacun des types d'interactions identifiés en amont section 2.3. Il s'est avéré que si l'IA modélisait légèrement mieux les composantes de L_{ij} et de C_{ij} , l'amélioration significative se trouvait au niveau du tenseur \mathbf{R} décrivant les interactions dites purement sous mailles. En effet, par définition le modèle LES est aveugle à ces interactions et les auteurs des modèles LES essaient au maximum d'enrichir ces derniers dans le but de simuler ces interactions et leurs conséquences. L'approche pilotée par les données en général et en particulier l'intelligence artificielle permet cet enrichissement de *manière induite, non supervisée* ; c'est-à-dire qu'il n'est plus nécessaire d'élaborer des formules

Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.91 / 0.72	0.91 / 0.72	0.91 / 0.71	0.96 / 0.87	0.91 / 0.72
$1 - R^2$	0.17 / 0.50	0.17 / 0.50	0.17 / 0.50	0.08 /0.28	0.17 /0.49
E_r	0.43 / 0.70	0.42 / 0.71	0.42 / 0.71	0.27 / 0.51	0.41 / 0.69

Table 5.18 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 61. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j} \tau_{ij}$ et des grandeurs scalaires Π et Ψ .

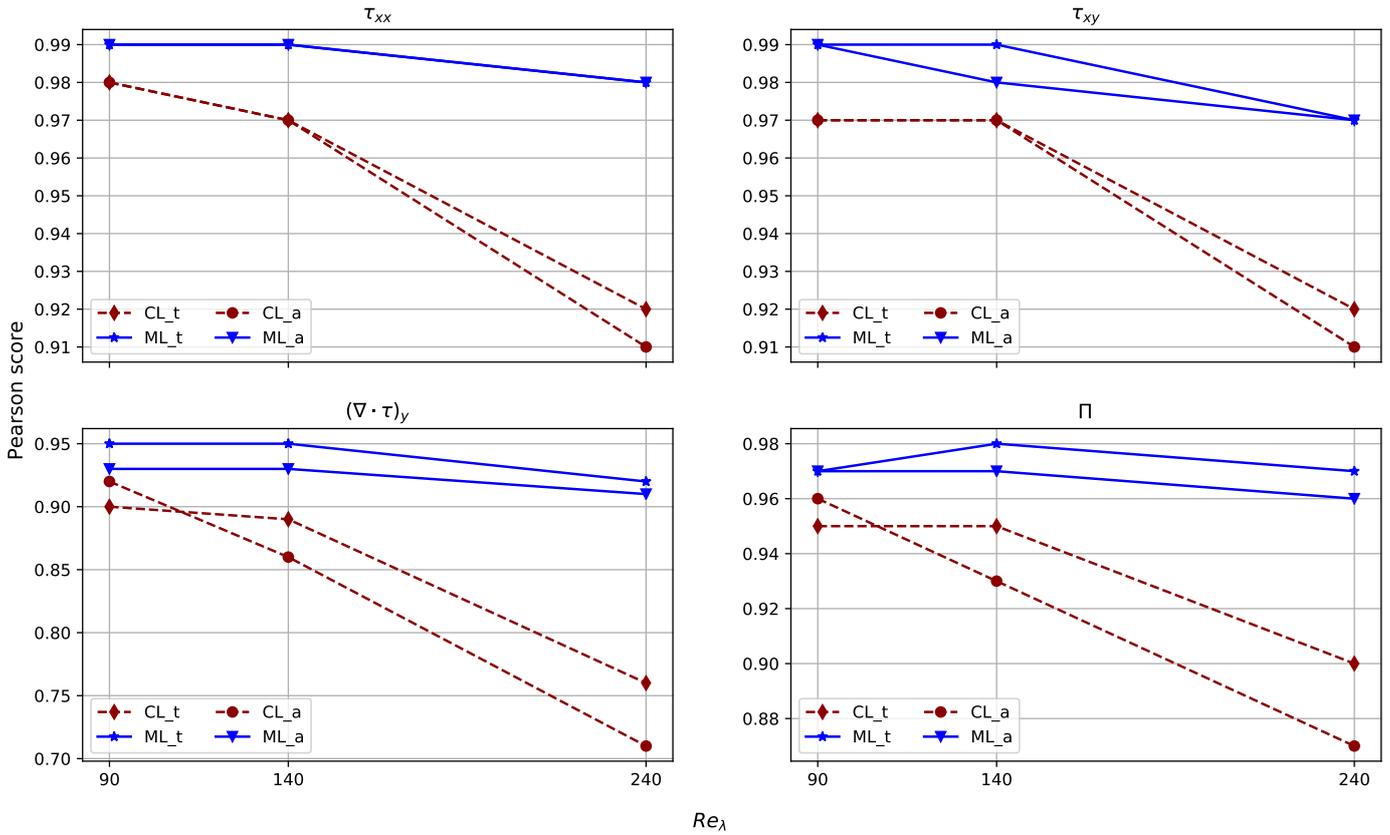


Figure 5.31 – Évolutions scores de Pearson en fonction du nombre de Reynolds. Les courbes solides sont les scores calculés en considérant la prédiction ; en pointillées sont obtenues en considérant les modélisation de Clark. L'indice a symbolise les scores pour l'itération avant le pic de dissipation ; t après l'établissement du régime développé de la turbulence

ou d'injecter des constantes pour tenir compte de telle ou telle réalité, l'IA *apprend* les nuances depuis les données directement. La modélisation n'est malgré tout pas parfaite, et il est possible qu'elle ne le sera jamais car les champs filtrés manquent *réellement* d'information.

Nous proposons comme fin de section 5.1 de représenter l'évolution des corrélations de Pearson au cours du temps pour plusieurs grandeurs : τ , $\nabla \cdot \tau$, Π , \mathbf{L} , \mathbf{C} et \mathbf{R} . Nous comparons les scores entre τ_{ij}^{ML} et les grandeurs construites à partir de ces composantes avec les scores considérant τ_{ij}^{Clark} et les grandeurs qui en découlent. Les figures sont tracées Fig.(5.33) et (5.32).

Nous avons considérés plusieurs itérations pour différents écoulements afin d'étudier les performances et l'étendue de ces dernières. Nous avons choisi les itérations pour représenter les différents régimes de la turbulence. Il est intéressant de résumer les résultats dans des figures présentant l'évolution des scores en fonction du Reynolds micro-échelle. Nous avons vu que pour le niveau tensoriel les performances sur les composantes diagonales étaient similaires, idem pour celles sur les composantes extradiagonales. De même au niveau vectoriel, les performances ne varient que très beau d'une composante à l'autre. Ce qui a été dit s'applique également aux composantes des tenseurs de la décompositions de Leonard. Ainsi, pour ne pas surcharger les figures, nous choisirons une de chacune de ces composantes. Pour continuer à marquer l'apport de nos travaux, nous renseignerons les performances obtenues par le modèle de Clark. Ces tracés

sont effectués figure Fig.(5.31).

Des visualisations 3D des grandeurs τ_{xx}^{FDNS} , τ_{xx}^{ML} et τ_{xx}^{Clark} sont également proposées figures Fig.(5.34). Elles ont été effectuées pour l'écoulement le plus turbulent considéré ($Re_\lambda \approx 240$) à l'itération $it = 61$ pour des raisons purement esthétiques. Elles permettent de voir à quoi correspondent les différences de corrélations entre modèles augmenté et algébrique. Au passage, on peut constater l'impact concret des décrochements aux queues considérés dans les PDFs : les événements rares, dont certains événements sont représentés en roses dans ces figures, n'apparaissent pas ou peu dans les contours de τ_{xx}^{Clark} , contrairement à ceux de τ_{xx}^{ML} très proches de ceux de la composante théorique.

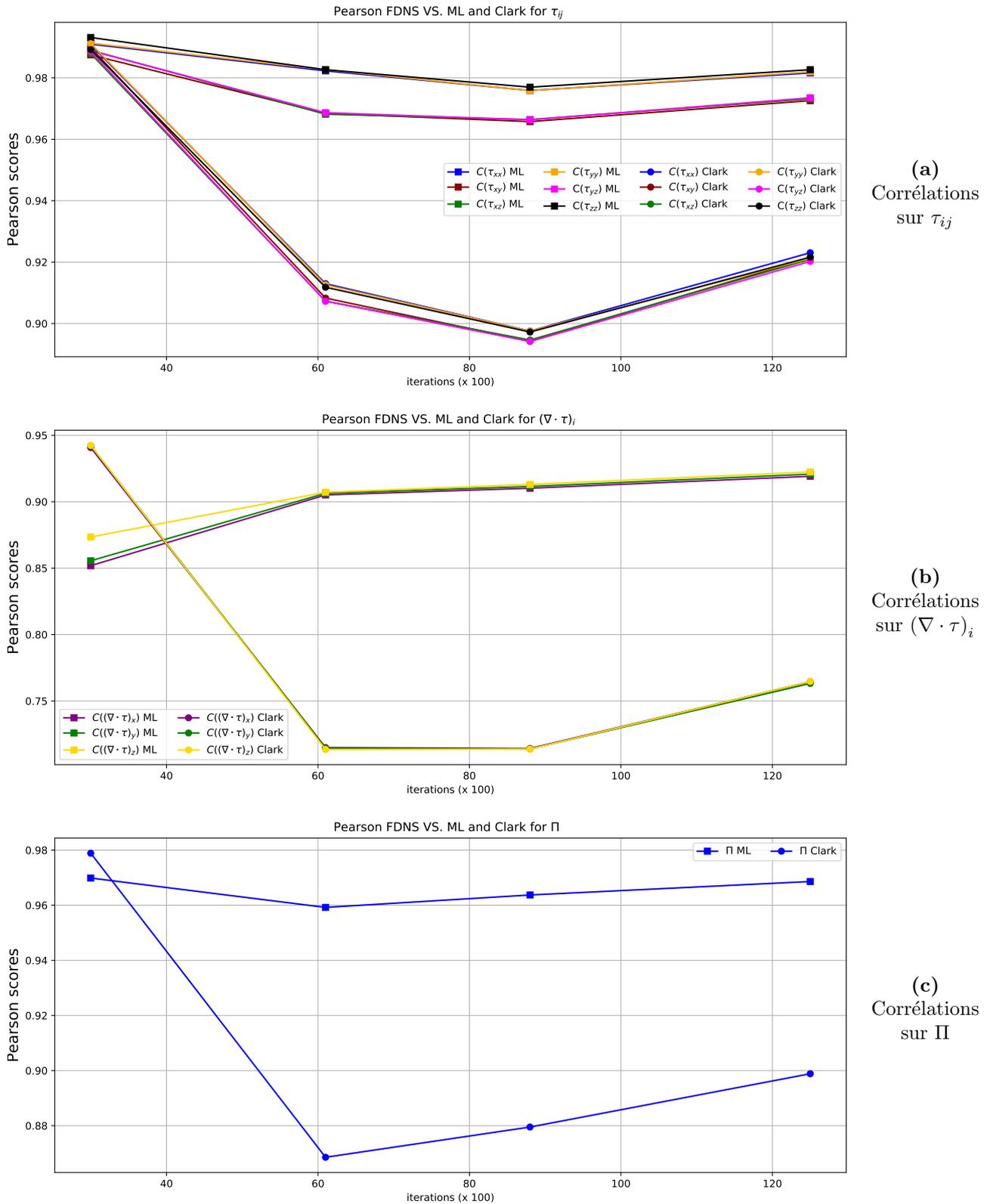


Figure 5.32 – Évolutions des corrélations FDNS vs. ML (courbes avec des **carrés**) et FDNS vs. Clark (courbes avec des **cercles**) aux niveaux tensoriel, vectoriel et scalaire.

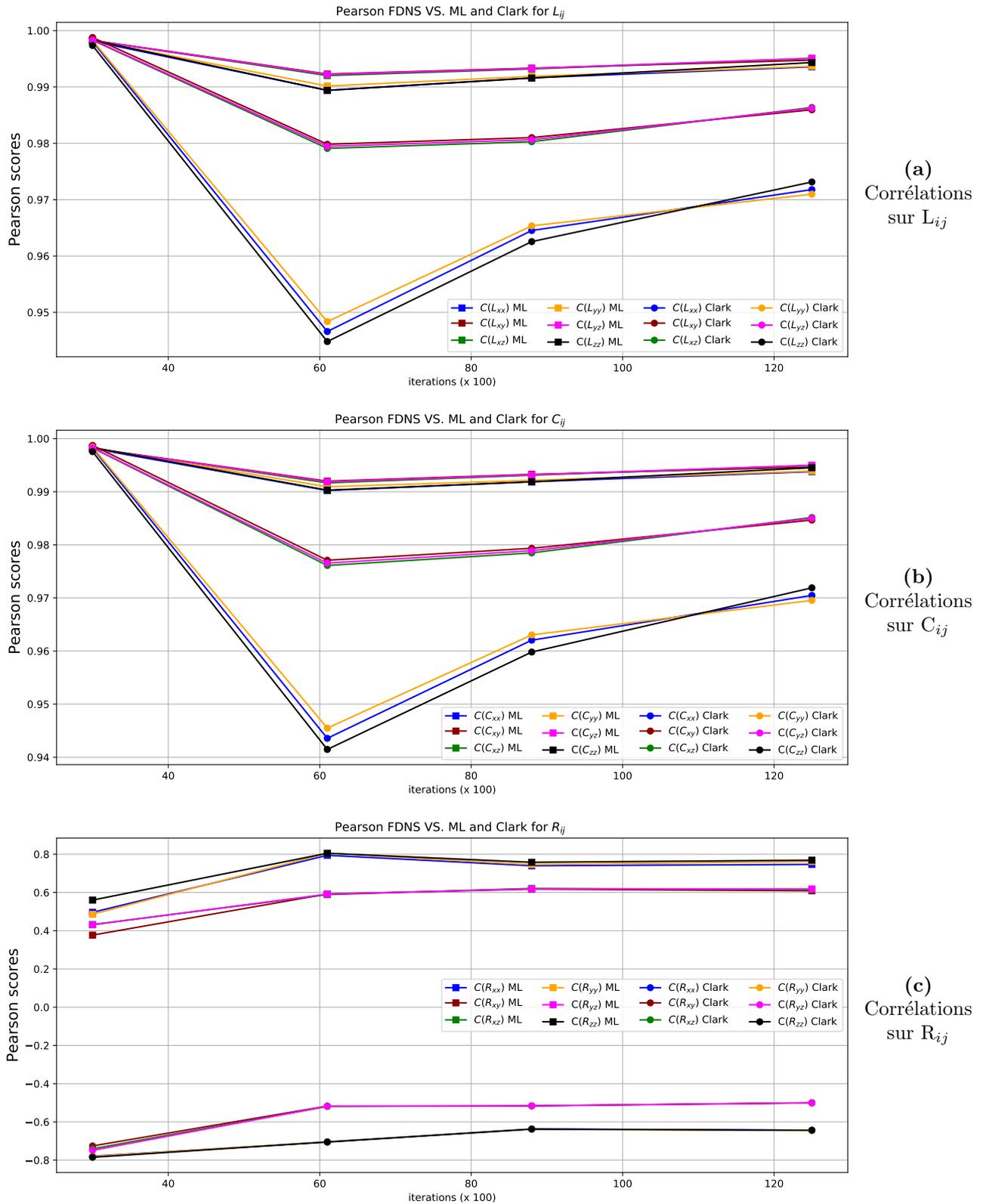
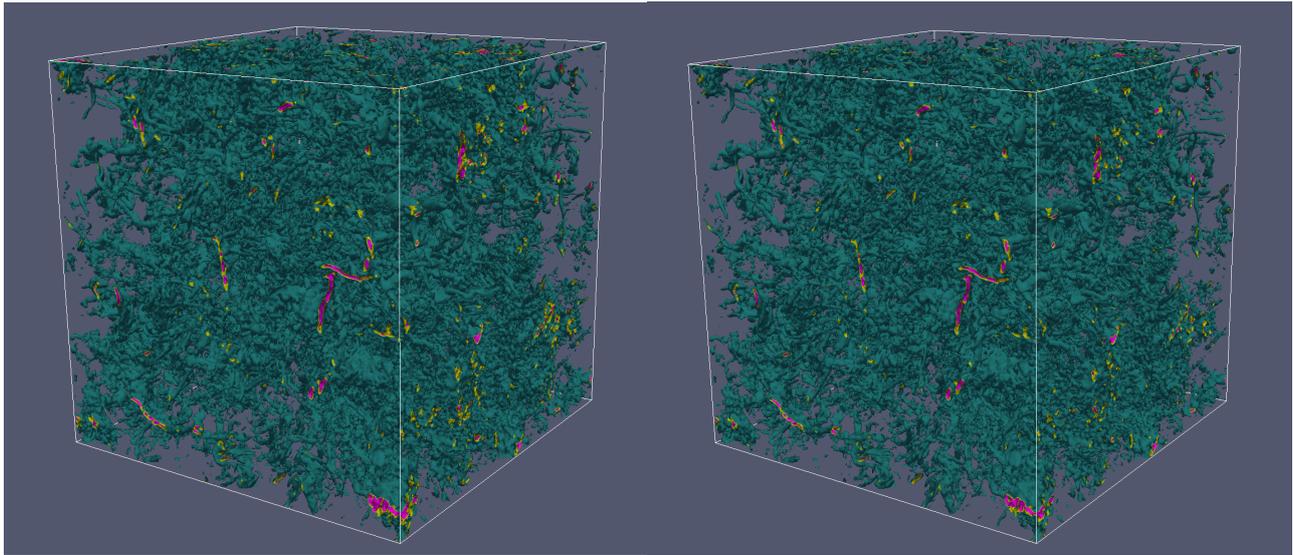
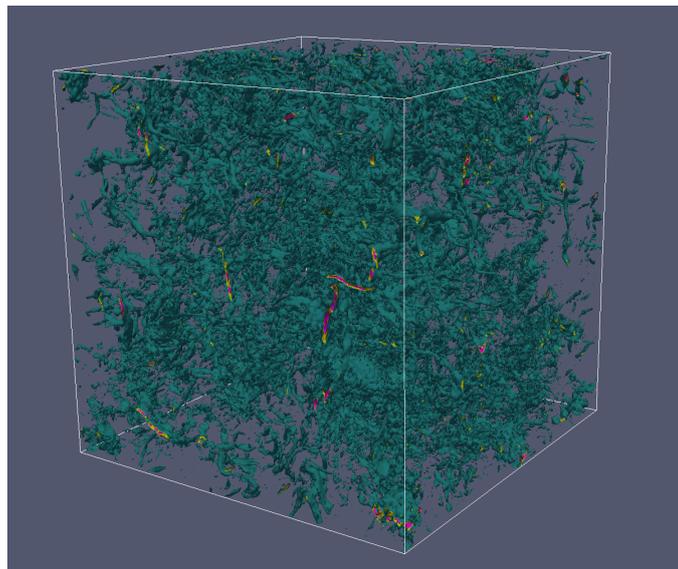


Figure 5.33 – Évolutions des corrélations FDNS vs. ML (courbes avec des carrés) et FDNS vs. Clark (courbes avec des cercles) des trois types d’interactions composant le tenseur τ conformément à la décomposition de Leonard (2.3.6).



(a) Contours de τ_{xx}^{FDNS}

(b) Contours de τ_{xx}^{ML}



(c) Contours de τ_{xx}^{Clark}

Figure 5.34 – En bleu iso 0.8 ; en jaune iso 0.13 ; en rouge iso 0.24. Le Reynolds est $Re_\lambda = 240$ et la résolution est de 256 points de discrétisation.

Coûts de calculs, parallélisation et construction d'un modèle LES - Il nous semble nécessaire de préciser une chose : la principale raison de l'utilisation de méthodes LES est le gain de temps que permettent ces modèles, quitte à perdre en information sur la Physique et donc en précision. La question du temps est fondamentale et au cœur même de la problématique des modèles LES. Dans le cas de l'IA on distingue le temps *d'entraînement*, et le temps de la *prédiction* générant la ou les grandeurs en fonction des entrées sélectionnées. Les deux sont indépendants.

Dans notre cas l'entraînement de l'IA, conformément à ce qui a été détaillé chapitre 4, a nécessité environ 12h de calcul *sur un seul GPU*. Les calculs des gradients au sein de chaque

noeud du réseau artificiel est une tâche facilement parallélisable et bien documentée. Au niveau de l'utilisation de ce réseau, la parallélisation est quasi totale puisque les cubes sont considérés comme complètement indépendants. Une infime partie du temps doit être considérée pour que chaque processeur communique sa prédiction au processeur principal et que ce dernier reconstruise le tenseur $\boldsymbol{\tau}$. C'est uniquement par manque de temps que nous n'avons pas exploré la parallélisation.

La majorité des prédictions ont été faites sur un ordinateur portable pour des raisons de praticité, sur un seul processeur CPU. Les temps de calculs ne sont donc absolument pas significatifs de ce qu'il est possible de faire même sans parallélisation.

Si notre modèle a une contrainte, il s'agit certainement de la nécessité de garder le facteur $R = 2$ (voir section 4.2.1). Or cette condition est équivalente à dire que $\kappa_C \neq \kappa_{\max}$ ce qui n'est pas intuitif avec la LES et à juste titre car lorsque l'on utilise une LES nous spécifions justement que $\kappa_C = \kappa_{\max}$. Pour utiliser le modèle que nous avons présenté, il faut alors considérer un autre type de LES conservant les dimensions physiques du problème. Les équations faisant apparaître τ_{ij} restent valables et le champ de vitesse est considéré comme filtré. Sous quelques aspects, le modèle qu'il faut développer ressemble aux modèles dynamiques qui nécessitent souvent de re-filtrer les champs de vitesse. Le terme convectif non linéaire n'apparaît donc pas et avec les moyens GPU actuels, il est très facile d'exploiter les forces de notre modèle tout en gagnant du temps par rapport à une résolution directe.

Chapitre 6

Conclusions des travaux et thèmes d'ouverture

Le fait que plusieurs échelles soient impliquées dans la description d'un phénomène de Turbulence impose des formulations mathématiques complexes qui aujourd'hui encore n'ont pas été totalement décodées. Numériquement, il est possible de résoudre ces équations temps par temps, mais pour de forts Reynolds cette tâche devient vite trop coûteuse en terme de temps de calculs, en particulier pour les applications industrielles. Il est nécessaire de réduire la gamme d'événements exactement résolus, complétant la dynamique par une modélisation des événements manquants. Dans ces nouveaux formalismes, le problème subsiste ouvert puisque des termes, inconnus, traduisant les interactions entre échelles résolues et non résolues apparaissent. Ils nécessitent la construction d'équations supplémentaires - de fermetures - liant les grandeurs disponibles aux quantités inconnues. Pendant plusieurs décennies des modèles abstraits (algébriques) ont été élaborés, et au fur et à mesure perfectionnés. Ces modèles permettent de prendre en compte des événements qui étaient invisibles aux modèles précédents, mais bien présents dans les données issues de simulations exactes et d'expériences. Les travaux qui ont été présentés tout au long de ce manuscrit s'inscrivent dans une dynamique récente et globale. Elle traduit la volonté de plus en plus marquée de la communauté des mécaniciens des fluides d'intégrer les solutions proposées par l'essor des intelligences artificielles, pour atteindre voire surpasser les performances de ces modèles algébriques.

Synthétiquement, nous pouvons dégager trois axes principaux au sein des différents chapitres de ce manuscrit. Premièrement, nous avons détaillé la structure d'un réseau de neurones et plus particulièrement des réseaux à convolutions. Nous avons situé chacun de ces éléments dans la trame générale de l'apprentissage, ainsi que ses objectifs. Nous avons également discuté des principes permettant d'élaborer une fonction de coût pouvant efficacement renseigner sur l'état des performances du réseau, avant de définir les caractéristiques, les forces et les faiblesses des principaux algorithmes de minimisation de cette fonction de coût. Ces éléments et en particulier les deux derniers sont capitaux dans l'apprentissage du réseau et assurent presque systématiquement des capacités de généralisation rivalisant avec celles de l'entraînement. Nous avons également détaillé la sélection des hyper-paramètres inhérents à toute intelligence artificielle qu'elle soit supervisée, renforcée ou non. Se basant sur des études issues d'une banque de documents en croissance quantitative et qualitative perpétuelle, ainsi que des milliers d'essais, nous avons présenté leurs rôles, impacts et propriétés (quand cela était nécessaire) pour

systematiser la démarche et l'analyse de leur optimisation. Nous soulignons le fait que l'utilisation optimale de l'IA nécessite ces concepts et ces recherches. Nous avons notamment proposé et étudié la fonction de coût $L_1L_{2\max}$ permettant de concentrer les erreurs à plusieurs échelles différentes et sur plusieurs amplitudes. Combinés avec la régularisation du réseau, les solutions obtenues tiennent en compte de toutes ces contraintes. Nous également présenté les fonctions d'activations Swish et Mish permettant des réseaux généralement plus performants que ceux impliquant les fonctions d'activations ReLU, Tanh ou Sigmoides, sans besoin de renormalisation après chaque convolution.

Deuxièmement, en nous appuyant sur des arguments physiques issus des travaux de Kraichnan, nous avons proposé un procédé de construction de base de données 3D systématique. Il a été montré qu'en choisissant la valeur du rapport R à partir de la taille du cube, nous pouvons assurer que la dynamique contenue dans le cube contribue à 92% des transferts de flux d'énergie à travers la coupure. La motivation derrière ce découpage est d'utiliser au maximum les données générées ou disponibles, et de façon indépendantes. Cette indépendance permet de réduire le nombre de simulations lancées ; dans notre cas, 10 itérations d'une seule simulation ont suffi pour entraîner une IA capable de généraliser son apprentissage sur des écoulements intrinsèquement différents. Si l'idée de découper le domaine pour conserver des données 3D n'est pas inédite¹, nous avons fourni un cadre dans lequel ce découpage était optimal s'appuyant sur la théorie de Nyquist et un ratio propre aux méthodes LES.

Un troisième et dernier axe concerne l'architecture utilisée et les résultats obtenus. Nous avons examiné le réseau U-net, ses caractéristiques et ses forces et expliqué en quoi il était le candidat idéal pour les travaux que nous entendions mener. Nous avons également trouvé appui dans des travaux de segmentation en Biomédical ou de prédictions de gradients en Combustion turbulente. Les performances sur les bases de données d'entraînement et de test furent très concluantes et très encourageantes notamment ceux des corrélations de plus de 99% entre champs totaux prédits et cibles, à partir d'entrées complètement inédites. En fonction de tous les travaux mentionnés dans les points précédents, il a été montré que conserver la structure des données et apprendre en volume pouvait amener à des résultats surpassant les attentes tant au niveau de l'apprentissage que de la généralisation, aussi bien du point de vue du Machine Learning (généralisation à des cas inédits) que de celui de la Mécanique des Fluides. En effet, nous avons établi que l'IA dont les poids ont été fixés lors de son apprentissage, pouvait fournir des résultats corrélés au sens de Pearson à plus de 99% malgré des champs de vitesses filtrés radicalement différents de l'ensemble d'apprentissage (voir par exemple les figures comparatives Figs.(5.20c)), dans leur nature ou dans leur degré de complexité. Mais pas seulement, il a été montré que cette performance se répercutait également aux niveaux vectoriel et scalaire avec des corrélations autour de 92% en moyenne contre 75% pour le modèle de Clark dans les cas les plus turbulents. Pour assurer une analyse prenant en compte différents aspects, nous avons considéré trois scores : celui de Pearson déjà évoqué, une mesure de la variance perdue par la prédiction (ou la modélisation) et enfin une erreur relative. Ils sont définis sur le volume global et selon les directions de l'espace. Cette façon d'analyser les résultats a permis d'en apporter un éclairage nouveau sur différents points :

- Il a été possible d'établir une isotropie dans l'apprentissage, affirmant alors que le réseau à convolutions est capable de presque parfaitement comprendre et situer chaque événement présent dans les champs d'entrée et de déduire les bonnes localisation et amplitude de

1. À la date de début de rédaction de ce manuscrit, deux articles (Beck et al. (2019) et Kurz and Beck (2020)) avaient utilisé ce procédé en parallèle de nos travaux (par rapport à leur date de publication)

leurs répercutions en tant que contrainte ;

- La mesure de la variance perdue a permis d'assurer que la quasi totalité des informations aux différents niveaux tensoriel, vectoriel et scalaire, a été retrouvée dans les champs prédits ou construits correspondants. L'erreur relative a fait apparaître le fait que le réseau est de manière absolue, plus précis sur les composantes diagonales que les composantes extra-diagonales dans les phases d'apprentissage et de prédiction.

Un des procédés que nous avons développé dans le chapitre 5 cherche à établir si nous pouvons expliquer de façon concrète la raison de la supériorité de la modélisation par IA sur celle du modèle de Clark. Pour se faire, nous avons étudié chacune des composantes de la décomposition de τ proposée par Leonard (1975) indépendamment les unes des autres. Il s'est avéré que la prédiction prenait en compte de la physique que la modélisation de Clark ne retrouvait pas ; notamment au niveau des interactions entre échelles non résolues. De plus, nous avons estimé que la rétrodiffusion de l'énergie cinétique des échelles non résolues vers les échelles résolues était globalement surestimée par le modèle de Clark contrairement au modèle augmenté qui prédit le bon pourcentage d'apparition de ce phénomène.

Une universalité supposée - Les tests du modèle sur des écoulements plus turbulents entendaient répondre (même partiellement) à une question : tout d'abord, est ce que la relation apprise entre le champ de vitesse filtré \bar{U} et le tenseur de contraintes τ est généralisable. Dans l'étendue de la gamme d'écoulements testés, puisque les performances globales et la précision des distributions obtenues sont similaires aux cas d'entraînement, nous répondons que oui. Une des raisons que nous pouvons évoquer qui expliquerait physiquement ce résultat est l'hypothèse d'universalité de la Turbulence isotrope de Kolmogorov. La relation apprise entre ces grandeurs est applicable même à d'autres cas, puisqu'il y a quelque chose d'universel qui lie les champs de vitesse de la Turbulence Isotrope. Cela implique que l'IA au travers de son architecture et son expressivité, a pu retrouver ce quelque chose d'universel et établir la relation latente entre \bar{U} et τ . Cette supposition demande à être vérifiée en appliquant ce modèle à des simulations plus turbulentes, et c'est ce que nous prévoyons de faire ; mais pour l'heure, les résultats des tests effectués semblent aller dans ce sens.

Ouverture de ces travaux

Ces travaux ouvrent plusieurs axes d'étude très variés. Nous en détaillerons trois :

- 1- l'élaboration d'un modèle LES sans hypothèse, dont la fermeture est donnée par un IA
- 2- savoir et comprendre jusqu'où la relation entre le champ de vitesse filtré et le tenseur de contraintes apprise sur ce jeu de données réduit s'étend
- 3- utilisation de réseaux encore plus sophistiqués, d'approches encore plus novatrices etc.

Chaînon manquant pour construire un modèle LES - Dans ces travaux, nous avons exploré en profondeur les performances *instantanées* du réseau. Pour agrémenter les résultats (tensoriels), nous avons estimé le devenir probable d'un modèle qui utiliserait l'U-net que nous avons entraînés. Ceci s'est fait en estimant les performances aux niveaux vectoriel et scalaire, et toujours en comparaison de modèles existants. La motivation est la suivante : si la prédiction et

les grandeurs dérivées sont plus précises au niveau instantané que les modèles de la littérature, c'est qu'a priori le modèle proposé est plus valable que les autres. Encore une fois, c'est l'étude des niveaux vectoriels et scalaires qui permettent d'établir cette conjecture. Il nous faut donc la tester en regardant l'évolution de la simulation LES augmentée en fonction du temps et comparer les spectres de vitesse obtenus sur le court, moyen voire long terme. Pour maximiser cette étude et aller plus loin, il est nécessaire d'étudier la parallélisation de notre framework, et d'utiliser au maximum l'indépendance des cubes pour justifier notre approche d'un point de vue concret. Il serait également intéressant d'appliquer ce modèle aux écoulements présentant de l'anisotropie, comme des obstacles. Dans un premier temps nous pouvons considérer deux IA : une telle que nous l'avons présenté, loin des obstacles ou des conditions aux limites particulières, et une autre tenant compte de l'anisotropie. Dans la première, les volumes peuvent être des cubes ; dans la deuxième il faudra considérer des volumes adaptés à la direction de l'anisotropie. Dans un second temps, il est possible de généraliser les types de volume en rajoutant des couches de padding pour faire coïncider les géométries différentes et résoudre indirectement la limite de non modulabilité des couches d'entrée dans les réseaux de neurones.

Applicabilité et universalité - Le point le plus essentiel de ces travaux est donc l'universalité de l'approche proposée ; de façon plus globale, c'est le cœur de tout travaux impliquant l'IA. La "scalabilité" de la parallélisation va être cruciale pour définir les limites de la généralisation. À ce moment là, il faudra rajouter des données dont le Re_λ est beaucoup plus fort, permettant de raffiner, de complexifier l'espace des solutions possibles. Selon notre compréhension, cet espace n'est pas à voir comme l'espace de fonctions des processus gaussiens. Il faut plutôt le voir comme des comportements supplémentaires à combiner, en fonction des patterns spécifiques que les écoulements beaucoup plus turbulents pourraient faire apparaître. Ce qui vient d'être dit s'applique aussi aux remarques du paragraphe précédent. Si la "scalabilité" est vérifiée, il sera envisageable d'utiliser ce genre de solutions pour des Reynolds très élevés, et/ou des écoulements présentant plus de complexité : champs magnétiques, forts cisaillements etc. Une des clés dans la compréhension de ces limites est l'interprétabilité de l'IA, domaine en plein essor. Il est possible de complexifier encore les réseaux à convolutions en imposant des filtres ou des poolings qui permettent une représentation des données dans des espaces de symétries de très haute dimensionnalité. L'idée est que les patterns appris sont invariants par rotation ou translation des entrées. Conformément à notre volonté d'utiliser la force brute de l'IA, les invariances ne sont pas induites par un travail sur les données ; elles sont déduites par le réseau de façon non supervisée.

Processus génératifs, transfert de style et autre - Des approches révolutionnaires se sont développées au cours des deux dernières années. Le processus génératif pur comme les GAN (Generative Adversarial Networks) font des prouesses dans l'art de générer des images, visages etc. dans leur domaine respectif. Ils ne sont pas directement applicables à la physique à cause du fait qu'à un ensemble de paramètres, la sortie n'est pas unique. La méthode très récente du StyleGan implique justement cette unicité et permet de générer de façon réversible une image à partir d'une autre. Par exemple, à partir d'une peinture, l'IA est capable de générer la même peinture mais dans le style de Van Gogh par exemple (d'où le nom Style Gan). Cette approche semble très adaptée à la génération des composantes du tenseur des contraintes ou tout autre grandeur turbulente à partir de grandeurs déjà définies. Les processus génératifs s'entraînent

à générer des données les plus fiables possibles, au point de tromper un “discriminateur”. On pourrait très bien imaginer un discriminateur qui prend en compte les invariances galiléennes ou d'autres contraintes purement physiques. Si en plus il y a correspondance exacte et réversibilité entre les entrées et la sortie attendue, nous sommes assurés de répondre à tous les critères et à toutes problématiques actuelles autour de la modélisation de grandeurs turbulentes.

Les processus génératifs offrent une autre particularité. Souvent, la génération nécessite l'apprentissage des distributions (au travers les valeurs moyennes et matrices de covariances, diagonales ou complètes) dépassant le cadre de l'estimation ponctuelle. Ainsi, il est possible de générer un champ et d'évaluer sa fiabilité.

Revenir aux fondamentaux pour aller plus loin - Initialement, notre modèle a été développé pour fermer le système d'équations de la LES. À partir des données qui décrivent au mieux la physique, l'IA se substitue aux phénomènes non considérés dans la LES. Pour aller plus loin, il est nécessaire de revenir aux bases de la MDF : la mécanique des milieux continus dont la philosophie a été abordée dans l'introduction. Un des axes que nous voulons développer avec les intelligences artificielles est de tenter de s'affranchir des hypothèses de continuité du milieu dans lequel les équations de la MDF s'appliquent. Plus précisément, nous voulons prédire des grandeurs macroscopiques à partir de la réalité sous-jacente au niveau de la molécule, à l'instar de la méthode de Boltzmann sur réseaux (Lattice Boltzmann Method, LBM)². Un commencement serait d'ailleurs le raffinement de la fonction d'équilibre autour de laquelle s'articule le passage mésoscopique-macroscopique de la LBM par IA. Nous pensons que c'est l'aboutissement de la logique même de l'établissement des théories : elles sont proposées, raffinées et selon les avancées scientifiques, corrigées. Nous pensons que l'IA peut être la voie liant tous les niveaux de la physique au delà des hypothèses qui fondent ces théories. Pour l'instant nous ne sommes qu'aux prémices, mais il est possible que réussissant à déchiffrer l'IA peut faire de cette dernière un formidable outil pour les théoriciens, expérimentalistes et numériciens, évidemment.

2. La LBM s'appuie sur une modélisation des équations de la physique statistique et plus particulièrement des équations de Boltzmann. Ces équations permettent d'obtenir l'évolution de la fonction de distribution des particules au sein d'un milieu. En prenant en compte des collisions entre particules et les relaxations dues au milieu dans lequel elles évoluent (comme la viscosité) après collision, ainsi que les densités de présence des particules, cette méthode permet de reconstruire des grandeurs macroscopiques utilisées en mécanique des fluides comme la quantité de mouvement, la densité (ou la masse) et d'obtenir leurs évolutions dans l'espace et dans le temps.

Bibliographie

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. (2016). Tensorflow : A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.
- Akolekar, H. D., Weatheritt, J., Hutchins, N., Sandberg, R. D., Laskowski, G., and Michellasi, V. (2019). Development and use of machine-learned algebraic reynolds stress models for enhanced prediction of wake mixing in low-pressure turbines. *Journal of Turbomachinery*, 141(4).
- Al Sayed, N. (2011). *Modèles LES invariants par groupes de symétries en écoulements turbulents anisothermes*. PhD thesis, Université de La Rochelle.
- Arjevani, Y., Shalev-Shwartz, S., and Shamir, O. (2015). On lower and upper bounds for smooth and strongly convex optimization problems. *arXiv preprint arXiv :1503.06833*.
- Aster, R. C., Borchers, B., and Thurber, C. H. (2011). *Parameter estimation and inverse problems*, volume 90. Academic Press.
- Balarac, G., Le Sommer, J., Meunier, X., and Volland, A. (2013). A dynamic regularized gradient model of the subgrid-scale scalar flux for large eddy simulations. *Physics of Fluids*, 25(7) :075107.
- Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.
- Bardina, J., Ferziger, J., and Reynolds, W. (1984). Improved turbulence models based on les of homogeneous incompressible turbulent flows. *Department of Mechanical Engineering, Report No. TF-19, Stanford*.
- Batchelor, G. (2000). *An introduction to fluid dynamics*. Cambridge university press.
- Beck, A., Flad, D., and Munz, C. (2019). Deep neural networks for data-driven turbulence models, arxiv. URL <http://arxiv.org/abs>.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
- Borue, V. and Orszag, S. A. (1998). Local energy flux and subgrid-scale statistics in three-dimensional turbulence. *Journal of Fluid Mechanics*, 366 :1–31.
- Brunton, S. L., Noack, B. R., and Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52 :477–508.
- Chollet, F. et al. (2015). Keras.
- Clark, R. A., Ferziger, J., and Reynolds, W. (1979). Evaluation of subgrid-scale models using an. *J. Fluid Mech*, 91(part 1) :1–16.
- Clevert, D.-A., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv :1511.07289*.
- Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

- Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2011). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1) :30–42.
- De Karman, T. and Howarth, L. (1938). On the statistical theory of isotropic turbulence. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 164(917) :192–215.
- Diakogiannis, F. I., Waldner, F., Caccetta, P., and Wu, C. (2020). Resunet-a : a deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162 :94–114.
- Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly notices of the royal astronomical society*, 450(2) :1441–1459.
- Dozat, T. (2016). Incorporating nesterov momentum into adam.
- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- Dumoulin, V. and Visin, F. (2018). A guide to convolution arithmetic for deep learning.
- Duraisamy, K. (2016). Informing turbulence closures with computational and experimental data. In *54th AIAA Aerospace Sciences Meeting*, page 1556.
- Duraisamy, K. (2020). Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. *arXiv preprint arXiv :2009.10675*.
- Duraisamy, K. and Durbin, P. (2014). Transition modeling using data driven approaches. In *Proceedings of the Summer Program*, page 427.
- Duraisamy, K., Iaccarino, G., and Xiao, H. (2019). Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51 :357–377.
- Duraisamy, K., Zhang, Z. J., and Singh, A. P. (2015). New approaches in turbulence and transition modeling using data-driven techniques. In *53rd AIAA Aerospace Sciences Meeting*, page 1284.
- Fang, R., Sondak, D., Protopapas, P., and Succi, S. (2018). Deep learning for turbulent channel flow. *arXiv preprint arXiv :1812.02241*.
- Ferziger, J. H., Perić, M., and Street, R. L. (2002). *Computational methods for fluid dynamics*, volume 3. Springer.
- Ferziger, R. C. J. and Reynolds, W. (1979). Evaluation of subgrid-scale models using an accurately simulated turbulent flow. *J. Fluid Mech*, 91 :1–16.
- Finlayson, S. G. (2020). *Learning Inductive Representations of Biomedical Data*. PhD thesis, Harvard University.
- Fletcher, R. (1987). *Practical methods of optimization*. John Wiley & Sons.
- Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation : Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.
- Gatski, T. B., Hussaini, M. Y., and Lumley, J. L. (1996). *Simulation and modeling of turbulent flows*. Oxford University Press.
- Gence, J. (1987). Le modèle de turbulence homogène et son utilité pour la compréhension du phénomène turbulent. *La Houille Blanche*, (7-8) :517–520.

- Gens, R. and Domingos, P. M. (2014). Deep symmetry networks. In *Advances in neural information processing systems*, pages 2537–2545.
- George, W. K. (2013). Lectures in turbulence for the 21st century. *Chalmers University of Technology*.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv :1406.2661*.
- Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., and Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789) :947–951.
- Haykin, S. S. et al. (2009). *Neural networks and learning machines/Simon Haykin*. New York : Prentice Hall,.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Hinze, J. (1975). Turbulence. *MacGraw Hill, New-York*.
- Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2004). Extreme learning machine : a new learning scheme of feedforward neural networks. In *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, volume 2, pages 985–990. Ieee.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization : Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv :1502.03167*.
- Jha, D., Smedsrud, P. H., Riegler, M. A., Johansen, D., De Lange, T., Halvorsen, P., and Johansen, H. D. (2019). Resunet++ : An advanced architecture for medical image segmentation. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 225–2255. IEEE.
- Kaandorp, M. L. and Dwight, R. P. (2018). Stochastic random forests with invariance for rans turbulence modelling. *arXiv preprint arXiv :1810.08794*.
- Kandi, H., Jain, A., Chathoth, S. V., Mishra, D., and Subrahmanyam, G. R. S. (2019). Incorporating rotational invariance in convolutional neural network architecture. *Pattern Analysis and Applications*, 22(3) :935–948.
- Kingma, D. P. and Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*.
- Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *Advances in Neural Information Processing Systems*, pages 971–980.
- Kolmogorov, A. N. (1941). The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. *Proceedings of the Royal Society of London. Series A : Mathematical and Physical Sciences*, 434(1890) :9–13.

- Kraichnan, R. H. (1976). Eddy viscosity in two and three dimensions. *Journal of the atmospheric sciences*, 33(8) :1521–1536.
- Krizhevsky, A. and Hinton, G. (2010). Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7).
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25 :1097–1105.
- Kurz, M. and Beck, A. (2020). A machine learning framework for les closure terms. *arXiv preprint arXiv :2010.03030*.
- Kutz, J. N. (2017). Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814 :1–4.
- Lakshminarayan, V. K. and Duraisamy, K. (2015). Adjoint-based estimation and control of spatial, temporal and stochastic approximation errors in unsteady flow simulations. *Computers & Fluids*, 121 :180–191.
- Landau, L. and Lifshitz, E. (1987). Theoretical physics, vol. 6, fluid mechanics.
- Lapeyre, C. J., Misdariis, A., Cazard, N., Veynante, D., and Poinso, T. (2019). Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203 :255–264.
- Lassen-Schmidt, B., Hering, A., Krass, S., and Meine, H. (2020). Automatic segmentation of the pulmonary lobes with a 3d u-net and optimized loss function. *arXiv preprint arXiv :2006.00083*.
- Layton, W. J. and Rebholz, L. G. (2012). *Approximate deconvolution models of turbulence : analysis, phenomenology and numerical analysis*, volume 2042. Springer Science & Business Media.
- Leonard, A. (1975). Energy cascade in large-eddy simulations of turbulent fluid flows. In *Advances in geophysics*, volume 18, pages 237–248. Elsevier.
- Li, J., Yang, Z., Liu, H., and Cai, D. (2018). Deep rotation equivariant network. *Neurocomputing*, 290 :26–33.
- Ling, J., Jones, R., and Templeton, J. (2016a). Machine learning strategies for systems with invariance properties. *Journal of Computational Physics*, 318 :22–35.
- Ling, J., Kurzawski, A., and Templeton, J. (2016b). Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807 :155–166.
- Liu, Q., Hang, R., Song, H., and Li, Z. (2017). Learning multiscale deep features for high-resolution satellite image scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 56(1) :117–126.
- Liu, S., Meneveau, C., and Katz, J. (1994). On the properties of similarity subgrid-scale models as deduced from measurements in a turbulent jet. *Journal of Fluid Mechanics*, 275 :83–119.
- Liu, Y., Ponce, C., Brunton, S. L., and Kutz, J. N. (2020). Multiresolution convolutional autoencoders. *arXiv preprint arXiv :2004.04946*.
- Liu, Y., Zhang, J., Gao, C., Qu, J., and Ji, L. (2019). Natural-logarithm-rectified activation function in convolutional neural networks. In *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, pages 2000–2008. IEEE.
- Love, M. (1980). Subgrid modelling studies with burgers' equation. *Journal of Fluid Mechanics*, 100(1) :87–110.

- Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv :1511.05644*.
- Mattila, K. (2010). Implementation techniques for the lattice boltzmann method. *Jyväskylä studies in computing*, (117).
- Maulik, R. and San, O. (2017). A neural network approach for the blind deconvolution of turbulent flows. *Journal of Fluid Mechanics*, 831 :151–181.
- Maulik, R., San, O., Rasheed, A., and Vedula, P. (2019). Subgrid modelling for two-dimensional turbulence using neural networks. *Journal of Fluid Mechanics*, 858 :122–144.
- Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv :1411.1784*.
- Misra, D. (2019). Mish : A self regularized non-monotonic neural activation function. *arXiv preprint arXiv :1908.08681*.
- Mohamed, A.-r., Dahl, G. E., and Hinton, G. (2011). Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1) :14–22.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- Mons, V., Chassaing, J.-C., Gomez, T., and Sagaut, P. (2016). Reconstruction of unsteady viscous flows using data assimilation schemes. *Journal of Computational Physics*, 316 :255–280.
- Murray, N. and Perronnin, F. (2014). Generalized max pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2473–2480.
- Nesterov, Y. (1983). A method solving a convex programming problem with convergence rate $o(1/k^2)$.
- Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- Pal, A. (2019). Deep learning parameterization of subgrid scales in wall-bounded turbulent flows. *arXiv preprint arXiv :1905.12765*.
- Pan, S. and Duraisamy, K. (2018). Long-time predictive modeling of nonlinear dynamical systems using neural networks. *arXiv preprint arXiv :1805.12547*.
- Parish, E. J. and Duraisamy, K. (2016). A paradigm for data-driven predictive modeling using field inversion and machine learning. *Journal of Computational Physics*, 305 :758–774.
- Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5) :1–17.
- Pope, S. (1975). A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(2) :331–340.
- Pope, S. B. (2001). *Turbulent flows*. IOP Publishing.
- Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Swish : a self-gated activation function. *arXiv preprint arXiv :1710.05941*, 7.
- Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT press Cambridge.

- Reina, G. A. and Panchumarthy, R. (2018). Adverse effects of image tiling on convolutional neural networks. In *International MICCAI Brainlesion Workshop*, pages 25–36. Springer.
- Richardson, L. F. (1992). *Weather prediction by numerical process*. Cambridge university press.
- Ronneberger, O., P.Fischer, and Brox, T. (2015). U-net : Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer. (available on arXiv :1505.04597 [cs.CV]).
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv :1609.04747*.
- Sagaut, P. (2006). *Large eddy simulation for incompressible flows : an introduction*. Springer Science & Business Media.
- Santos, C. d., Tan, M., Xiang, B., and Zhou, B. (2016). Attentive pooling networks. *arXiv preprint arXiv :1602.03609*.
- Sarghini, F., De Felice, G., and Santini, S. (2003). Neural networks based subgrid scale modeling in large eddy simulations. *Computers & fluids*, 32(1) :97–108.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv :1312.6120*.
- Scherer, D., Müller, A., and Behnke, S. (2010). Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer.
- Schmidhuber, J. (2015). Deep learning in neural networks : An overview. *Neural networks*, 61 :85–117.
- Schumann, U. (1991). Subgrid length-scales for large-eddy simulation of stratified turbulence. *Theoretical and Computational Fluid Dynamics*, 2(5) :279–290.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.
- Singh, A. P. and Duraisamy, K. (2016). Using field inversion to quantify functional errors in turbulence closures. *Physics of Fluids*, 28(4) :045110.
- Singh, A. P., Duraisamy, K., and Zhang, Z. J. (2017a). Augmentation of turbulence models using field inversion and machine learning. In *55th AIAA Aerospace Sciences Meeting*, page 0993.
- Singh, A. P., Medida, S., and Duraisamy, K. (2017b). Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA Journal*.
- Smagorinsky, J. (1963). General circulation experiments with the primitive equations : I. the basic experiment. *Monthly weather review*, 91(3) :99–164.
- Spalart, P. and Allmaras, S. (1992). A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439.
- Staudemeyer, R. C. and Morris, E. R. (2019). Understanding lstm—a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv :1909.09586*.
- Sugiyama, M. and Kawanabe, M. (2012). *Machine learning in non-stationary environments : Introduction to covariate shift adaptation*. MIT press.
- Sutskever, I. (2013). *Training recurrent neural networks*. PhD thesis.

- Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR.
- Sutton, R. (1986). Two problems with back propagation and other steepest descent learning procedures for networks. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986*, pages 823–832.
- Taghanaki, S. A., Bentaieb, A., Sharma, A., Zhou, S. K., Zheng, Y., Georgescu, B., Sharma, P., Xu, Z., Comaniciu, D., and Hamarneh, G. (2019). Select, attend, and transfer : Light, learnable skip connections. In *International Workshop on Machine Learning in Medical Imaging*, pages 417–425. Springer.
- Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.
- Tato, A. and Nkambou, R. (2018). Improving adam optimizer.
- Teaca, B., Verma, M., Knaepen, B., and Carati, D. (2009). Energy transfer in anisotropic magnetohydrodynamic turbulence. *Physical Review E*, 79(4) :046312.
- Tieleman, T. and Hinton, G. (2012). Divide the gradient by a running average of its recent magnitude. coursera neural network lecture. 6 :26–31.
- Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method. *arXiv preprint physics/0004057*.
- Tran, C. V. (2009). The number of degrees of freedom of three-dimensional navier–stokes turbulence. *Physics of Fluids*, 21(12) :125103.
- Vollant, A. (2015). *Evaluation et développement de modèles sous-maille pour la simulation des grandes échelles du mélange turbulent basés sur l’estimation optimale et l’apprentissage supervisé*. PhD thesis.
- Vreman, B., Geurts, B., and Kuerten, H. (1996). Large-eddy simulation of the temporal mixing layer using the clark model. *Theoretical and Computational Fluid Dynamics*, 8(4) :309–324.
- Vreman, B., Geurts, B., and Kuerten, H. (1997). Large-eddy simulation of the turbulent mixing layer. *Journal of fluid mechanics*, 339 :357–390.
- Wang, J.-X., Wu, J., Ling, J., Iaccarino, G., and Xiao, H. (2017). A comprehensive physics-informed machine learning framework for predictive turbulence modeling. *arXiv preprint arXiv :1701.07102*.
- Wilcox, D. C. et al. (1998). *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA.
- Wu, J.-L., Xiao, H., and Paterson, E. (2018). Data-driven augmentation of turbulence models with physics-informed machine learning. *arXiv preprint arXiv :1801.02762*.
- Xiao, H. and Cinnella, P. (2019). Quantification of model uncertainty in rans simulations : A review. *Progress in Aerospace Sciences*, 108 :1–31.
- Xie, C., Wang, J., Li, H., Wan, M., and Chen, S. (2019). Artificial neural network mixed model for large eddy simulation of compressible isotropic turbulence. *Physics of Fluids*, 31(8) :085112.
- Yuan, Z., Xie, C., and Wang, J. (2020). Deconvolutional artificial neural network models for large eddy simulation of turbulence. *Physics of Fluids*, 32(11) :115106.
- Zeiler, M. D. (2012). Adadelta : An adaptive learning rate method.

- Zhang, R. (2019). Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, pages 7324–7334. PMLR.
- Zhang, Z., Liu, Q., and Wang, Y. (2018). Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5) :749–753.
- Zhou, Z., He, G., Wang, S., and Jin, G. (2019). Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network. *Computers & Fluids*, 195 :104319.
- Zhu, L., Zhang, W., Kou, J., and Liu, Y. (2019). Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Physics of Fluids*, 31(1) :015105.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1) :43–76.

Annexe A

Suppléments sur la minimisation d'une fonction de coût et le moment

Cette annexe entend apporter plus de détails sur l'élaboration des méthodes d'optimisation des réseaux de neurones modernes comme les algorithmes Adam et Nadam Algo.(5) et (6). Nous apportons quelques éléments théoriques sur l'optimisation des réseaux de neurones.

A.1 Limite de la descente de gradient basique

En utilisant les notations du chapitre 3, la fonction de coût se note \mathcal{J} , elle est définie sur un ensemble de paramètres que nous avons représentés par θ . Dans la section concernant les algorithmes de minimisation nous avons noté le gradient de la fonction de coût vis-à-vis des paramètres $\nabla = \partial_{\theta} \mathcal{J}(\theta)$ ou $g_i = \partial_{\theta_i} \mathcal{J}(\theta)$. On définit également la Hessienne ou dérivée seconde de la fonction de coût comme la variation des gradients de \mathcal{J} par rapport aux paramètres. Elle traduit la courbure de la fonction de coût sur l'espace des paramètres Θ . On l'écrit

$$H_{i,j} = \frac{\partial g_i}{\partial \theta_j} \quad (\text{A.1.1})$$

Elle apparaît lorsque l'on considère le développement de Taylor de la fonction de coût autour de θ au deuxième ordre, avec η le taux d'apprentissage (Goodfellow et al., 2016, chap. 4). On remanie ce développement pour exprimer l'évolution de la fonction de coût entre deux points :

$$\mathcal{J}(\theta = \theta_0 - \eta \mathbf{g}) - \mathcal{J}(\theta_0) \approx - \underbrace{\eta \mathbf{g}^T \mathbf{g}}_{\text{Variations linéaires}} + \underbrace{\frac{1}{2} \eta^2 \mathbf{g}^T \mathbf{H} \mathbf{g}}_{\text{Correction liée à la courbure}} \quad (\text{A.1.2})$$

Le terme au premier ordre représente la variation linéaire due à la présence du gradient. La direction dans l'espace des paramètres est donnée par le vecteur unitaire

$$\mathbf{p} = -\frac{\mathbf{g}}{\mathbf{g}^T \mathbf{g}} = -\frac{\mathbf{g}}{\|\mathbf{g}\|} \quad (\text{A.1.3})$$

Algorithm 7: Descente de gradient dans sa version basique (SGD)

```

initialiser poids  $\theta_0$  ;
initialiser taux d'apprentissage  $\eta$  ;

while  $t \leq T$  do
  |  $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$  ;
  |  $\theta_t \leftarrow \theta_{t-1} - \eta g_t$ 
end

```

Le taux d'apprentissage constitue le pas avec lequel l'optimisation parcourt l'espace des paramètres d'un point à l'autre. Le deuxième terme représente les effets de courbure par rapport à la direction que l'algorithme semble emprunté. Les effets de courbures sont très importants en théorie classique de l'optimisation et l'algorithme de descente de gradient¹ y est aveugle. Les détails de l'algorithme sont rappelés Algo.(1). Par exemple, la condition d'Armijo imposée dans le BFGS (Nocedal and Wright, 2006) impose une valeur seuil de perte d'erreur d'un pas à l'autre. Or il s'avère que

$$\frac{\|g\|}{g^T H g} \ll 1 \Rightarrow \mathcal{J}(\theta) - \mathcal{J}(\theta_0) > 0 \quad (\text{A.1.4})$$

menant à la divergence de l'optimisation. A priori donc, la courbure est nécessaire pour envisager une minimisation de la fonction de coût efficace. Enfin, dans des zones de minimum locaux où la dérivée est nécessairement faible, un grand pas est nécessaire pour continuer l'apprentissage et inversement, dans des zones de forts gradients de petits pas sont souvent nécessaires pour gagner en stabilité, voir Fig.(A.1). Dans un réseau de neurones par exemple, pour optimiser la découverte de nouveaux patterns ou caractéristiques de l'entrée il est nécessaire que l'ajustement des poids n'interfèrent pas avec les patterns déjà découverts, ainsi il est recommandé de réduire les pas dans les directions où les gradients sont forts c'est-à-dire là où le réseau a déjà appris, et se concentrer sur d'autres sous espaces de l'espace des paramètres (Sutton, 1986).

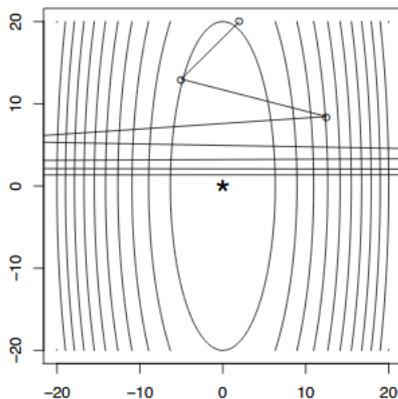


Figure A.1 – Illustration de la divergence due à une mauvaise évaluation du pas de recherche dans l'espace des paramètres et de l'absence de terme compensant les oscillations.

La problématique est alors la suivante : les problèmes d'apprentissage comptent un nombre de paramètres élevé, et c'est encore plus vrai pour le deep learning qui compte dans la majorité

1. Nature ou vanillé selon les langues.

des cas des millions de paramètres. Il est donc inconcevable d’imaginer calculer la hessienne de la fonction de coût, dans la mesure où l’on veut atteindre un résultat. Nous pouvons étendre cet argument à toutes les méthodes d’apprentissage du second ordre ou d’approximations du second ordre comme le BFGS (Nocedal and Wright, 2006). D’un autre côté, l’utilisation de la descente de gradient tel quelle n’est pas non plus envisageable car le manque d’information sur la courbure fait rapidement diverger l’optimisation. La parade artificielle de considérer des taux d’apprentissages très petits de l’ordre du millionième n’est pas valable car cela ralentit drastiquement l’apprentissage et augmente les risques de stagnation dans un minimum local ou gradient évanescent bloquant complètement l’apprentissage. L’absence de stratégie de *line search* bien définie pour établir la taille du pas dans Θ , rend très difficile l’atteinte du minimum global (c’est-à-dire un minimum local satisfaisant le critère de convergence). Utiliser le SGD s’accompagne obligatoirement avec une planification de la réduction du taux d’apprentissage réduisant les plus grosses oscillations comme celles présentent dans la figure (A.1). Cependant planifier une décroissance (continue, plateau, exponentiellement etc) du taux d’apprentissage n’est pas véritablement dans la logique du *line search* puisque sa diminution dépendra de la vitesse de diminution de la fonction de coût et non de l’état de certains poids ou des directions à privilégier.

Les optimiseurs développés par la suite dont certains ont été abordés chapitre 3 enrichissent l’information disponible sur le gradient renseignant sur la courbure sous-jacente de la fonction de coût entre autre. Intuitivement, une étape supplémentaire est rajoutée pour évaluer l’inertie de l’apprentissage selon toutes les directions possibles. L’idée est d’encourager les directions où les gradients sont les plus consistants. Cela évite les oscillations comme celles Fig.(A.1) qui peuvent amener à la divergence. Il existe deux façons de réaliser cette opération. La section suivante amène plus de détails sur ces procédés appelés moment (classique) ou moment de Nesterov.

A.2 Utilisation du moment en substitution à la Hessienne : moment classique et de Nesterov

Moment classique - On rappelle l’algorithme du moment classique (abrégié CM pour Classical Momentum) (Polyak, 1964) :

Algorithm 8: Descente de gradient avec moment

```

initialiser poids  $\theta_0$  ;
initialiser taux d’apprentissage  $\eta$  ;
initialiser constante de déclin  $\mu$  ;
initialiser  $m_t = 0$  ;

```

```

while  $t \leq T$  do

```

```

     $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1})$  ;
     $m_t \leftarrow \mu m_{t-1} + \eta g_t$  ;
     $\theta_t \leftarrow \theta_{t-1} - m_t$  ;

```

```

end

```

Le moment apparaît comme un terme qui garde en mémoire les précédents gradients. Le coefficient qui détermine l'impact de cette mémoire est généralement dénoté μ . L'“oubli” d'un gradient est exponentiel². Au fur et à mesure, l'optimiseur ainsi construit accumule les directions; si les gradients accumulés pointent vers une direction donnée, alors l'optimiseur encouragera d'autant plus cette direction. Au contraire si les gradients pointent vers des directions différentes, alors l'optimiseur considérera une direction intermédiaire. Cette méthode donne ainsi à l'optimiseur un contexte par rapport à la décision au temps t et permet de converger plus rapidement que la méthode basique. Les contributions des gradients constituent également une forme de calcul de pas corrigeant les principaux défauts du SGD; cela permet également d'éviter les directions où le gradient oscille (Dozat, 2016). Par construction, le CM est plus actif lorsque les valeurs propres de la Hessienne sont proches (zones de courbures faibles) les unes des autres, c'est-à-dire lorsque la courbure est sensiblement la même dans toutes les directions (Goodfellow et al., 2016). Il existe des schémas d'initialisation du moment : le plus basique consiste à l'initialiser à 0.9 ou plus généralement proche de 1 (Goodfellow et al., 2016). Hinton (lecture 6c de ses enseignements) propose de l'initialiser à 0.5 puisque durant les premières itérations le moment n'apparaît pas comme indispensable. Puis lorsque le learning avec l'algorithme basique semble ralentir, Hinton augmente cette valeur à 0.9 et ultimement à 0.99. De ce fait, l'optimiseur pourra éviter ou sortir d'une région dont les paramètres constituent un minimum local de J , ou du moins amplifie les gradients des régions (dans lesquelles l'optimiseur ne semble pas pouvoir sortir) ou la courbure est faible, ceci en vue d'augmenter les performances de l'algorithme (Sutskever et al., 2013). Il est alors clair cette méthode informe directement sur une courbure sous jacente, sans pour autant la calculer et encore moins l'inverser (voir également à ce sujet (Sutskever, 2013, chap.2) sur les moments.

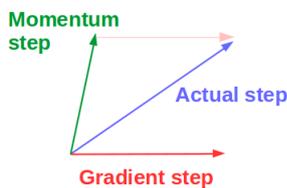


Figure A.2 – Illustration de l'action du moment sur la direction dans l'espace des paramètres que l'optimiseur va finalement emprunter. Un cas limite apparaît lorsque la courbe verte (direction m_t) pointe dans un mauvais sens ou dans le sens opposé de la correction proposée par le gradient (en rouge g_t).

Il s'avère parfois que le moment pointe vers une direction opposé au gradient Ruder (2016). Ceci peut être visualisé à partir de la figure Fig.(A.2). Puisque la direction finale (flèche bleu) est la composition des direction donnée par le gradient (rouge) et par le moment (rouge), dès lors que pour une raison ou une autre le moment pointe vers une mauvaise direction, l'optimiseur est alors contre-productif et peut revenir en arrière. Pour stabiliser le moment Nesterov (1983) proposa une version accéléré appelée Moment Accéléré de Nesterov (NAG).

Moment accéléré de Nesterov (NAG) - Écrivons l'algorithme de NAG pour mieux en comprendre la différence subtile entre le NAG et le CM :

2. On peut le voir en écrivant m_{t-1} comme le résultat de la somme d'éléments impliquant $t-2$ qui ramené à la formule de m_t apparaîtra comme pondéré par μ^2 et ainsi de suite.

Algorithm 9: Descente de gradient avec moment accéléré de Nesterov (Nesterov, 1983)

```

initialiser poids  $\theta_0$  et  $m_t = 0$  ;
initialiser taux d'apprentissage  $\eta$  et la constante de déclin  $\mu$  ;
while  $t \leq T$  do
     $g_t \leftarrow \nabla_{\theta_{t-1}} f(\theta_{t-1} - \mu m_{t-1})$  ;
     $m_t \leftarrow \mu m_{t-1} + g_t$  ;
     $\theta_t \leftarrow \theta_{t-1} - \eta m_t$  ;
end

```

Nous reconnaissons les deux dernières étapes qui sont le calcul du moment par accumulations des gradients précédents et l'ajustement des poids selon la direction proposée. En regardant attentivement, on remarque que ce n'est pas exactement le gradient des paramètres qui est calculé mais le gradient des paramètres tels qu'ils auraient été si le moment avait été appliqué pour l'ajustement des poids. C'est là la force du Nesterov par rapport au CM (Sutskever et al. (2013), Sutskever (2013), Ruder (2016)). L'intuition derrière cette projection dans un futur possible est que le gradient pointe toujours vers la direction de minimisation (peu importe le pas), y compris si le moment (classique) pointait vers une direction opposé à la direction de descente. De cette façon, le moment de Nesterov (qui n'en est pas vraiment un dans la version originale) permet de continuer dans la lignée des meilleures directions atténuant les forts gradients (directions) apparaissant dans le CM ou le SGD, assurant alors un apprentissage plus consistant et plus rapide. La méthode ainsi proposée corrige peu ou beaucoup l'erreur du moment à chaque itération (Sutskever, 2013). Enfin, des études (Sutskever et al., 2013) prouvent que le NAG peut être utile et efficace même lorsque les directions propres de la Hessienne sont fortes et différentes contrairement au CM voir également Goodfellow et al. (2016, chap.8) à ce sujet.

Comme nous l'avons mentionné aux chapitres 3 et 4, ce n'est que lorsque nous avons eu recours aux NAG dans l'algorithme Adam (Kingma and Ba, 2014) que nos modèles ont pu véritablement converger. C'est l'optimiseur Nadam élaboré par Dozat (2016). Rappelons que des explorations d'hyper-paramètres : le taux d'apprentissage, la profondeur du réseau, l'initialisation des poids, des valeurs des grandeurs μ ont été réalisées pour l'utilisation de Adam qui est le standard des travaux récents existants. Malgré tout, les runs utilisant l'algorithme Nadam se sont avérés plus performants et plus rapides. Le constat est d'ailleurs le même pour les travaux de prédictions des composantes des contraintes sous-mailles et les travaux de reconstructions du champ de vitesse chapitre, pour les architectures U-net et DenseNet³. Nous pouvons proposer la raison suivante : la courbure des fonctions de coût MAE ou L1L2 que nous avons essayé de minimiser avec Adam sont trop différentes en différents points de l'espace et qu'également elles présentent des minima locaux assez forts dont seul les propriétés du NAG permettent de s'extirper.

3. Nous ne présenterons pas ces travaux de reconstruction pour des raisons de temps mais ils seront l'objet d'un article. Idem, le DenseNet est une architecture qui utilise les réseaux à convolution ; nous n'avons pas eu l'occasion de présenter les résultats les utilisant

Annexe B

Plus de détails sur les résultats présentés Chapitre 5

B.1 Résultats complémentaires pour la simulation dont Reynolds $Re_\lambda = 140$

B.1.1 Figures supplémentaires pour l'itération 118

Nous traçons les PDFs sur la grandeur scalaire Ψ pour compléter l'analyse du régime développé de la simulation vérifiant $Re_\lambda = 140$. Les performances des deux modèles sont plutôt similaires, même si le scalaire construit à partir de τ^{ML} décroche moins aux queues des PDFs. Les figures Fig.(B.2) illustrent les erreurs relatives calculées le long des axes entre τ_{ij}^{FDNS} et τ_{ij}^{ML} figures Figs.(B.2a), (B.2c) et (B.2e) et entre τ_{ij}^{FDNS} et τ_{ij}^{Clark} figures Figs.(B.2b), (B.2d) et (B.2f).

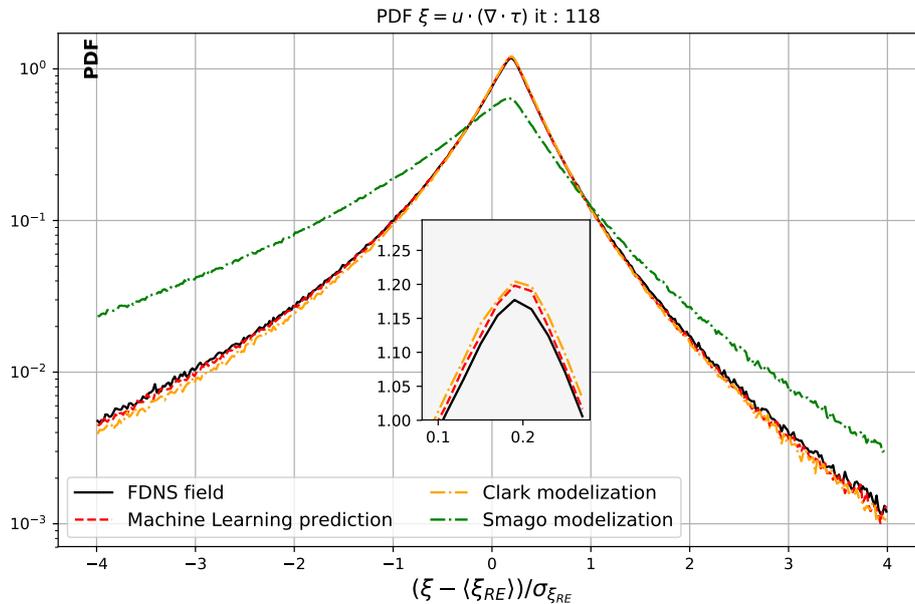


Figure B.1 – À l'itération 118, comparaisons des PDFs des champs scalaires Ψ^{FDNS} (courbe continue noire), avec Ψ^{ML} (ligne brisée rouge), Ψ^{Clark} et Ψ^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont standardisées par les moyennes et déviations de la FDNS.

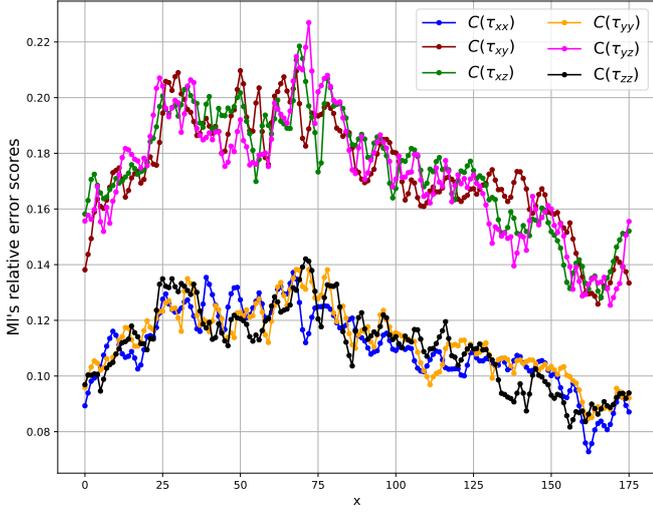
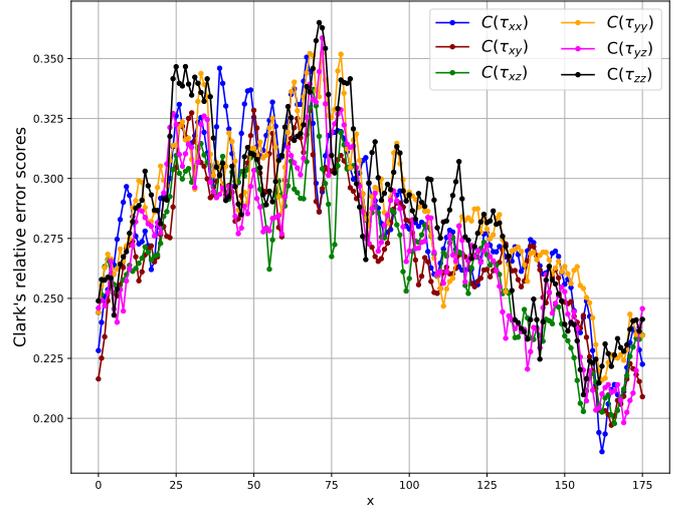
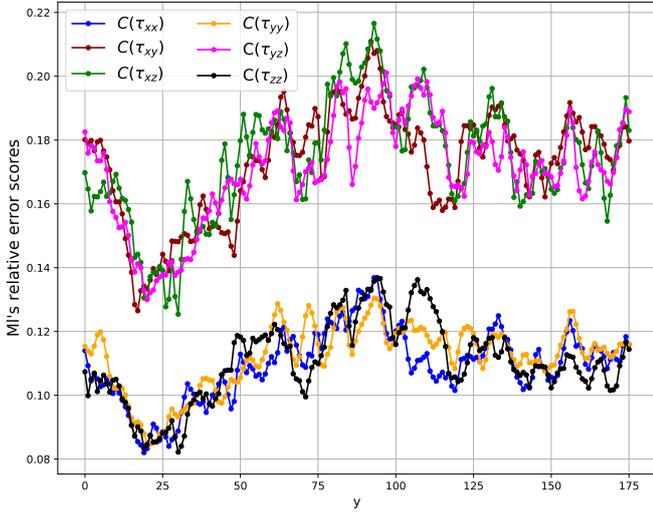
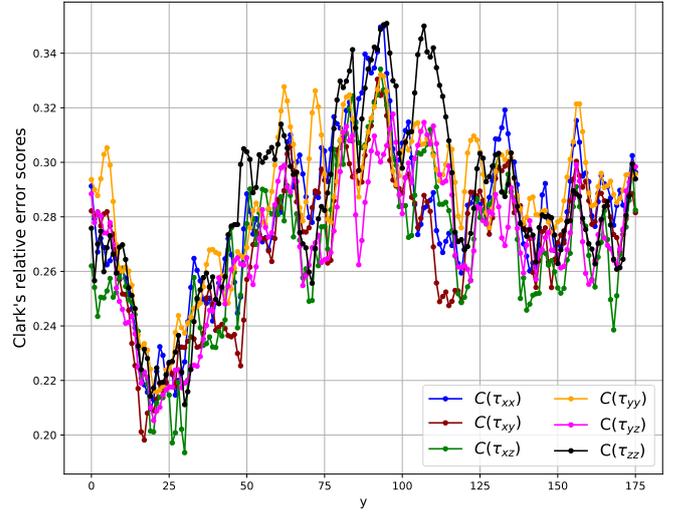
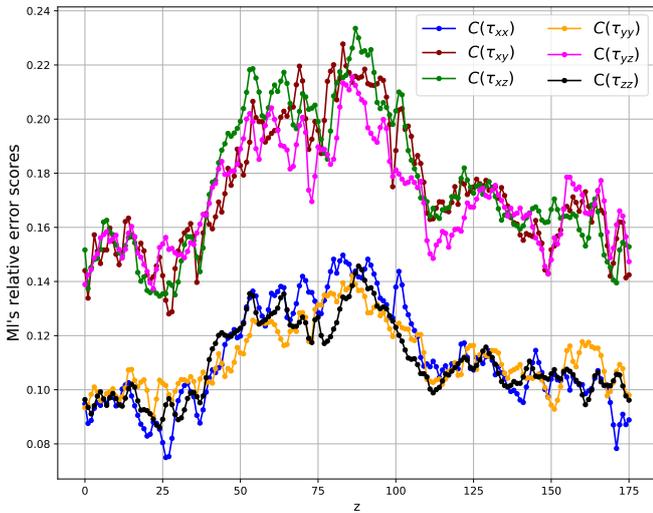
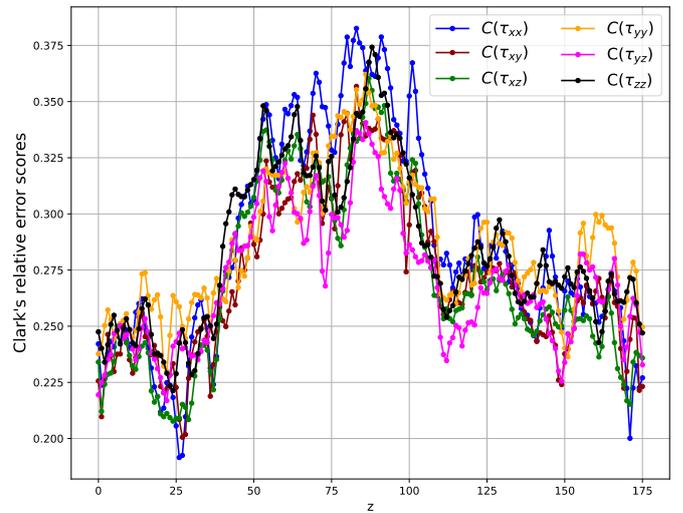
(a) Erreurs relatives le long de l'axe x (ML)(b) Erreurs relatives le long de l'axe x (Clark)(c) Erreurs relatives le long de l'axe y (ML)(d) Erreurs relatives le long de l'axe y (Clark)(e) Erreurs relatives le long de l'axe z (ML)(f) Erreurs relatives le long de l'axe z (Clark)

Figure B.2 – Erreurs relatives entre les composantes de τ_{ij}^{FDNS} et celles de τ_{ij}^{ML} (figures à gauche). Idem avec les composantes de τ_{ij}^{Clark} (figures à droite) à la place de la prédiction.

B.1.2 Figures supplémentaires pour l'itération 88

PDFs au niveau tensoriel - En complément des résultats présentés section (5.1.2), nous traçons les comparaisons des PDFs issus des composantes τ_{ij}^{FDNS} , τ_{ij}^{ML} , τ_{ij}^{Clark} et τ_{ij}^{Smago} Fig.(B.5).

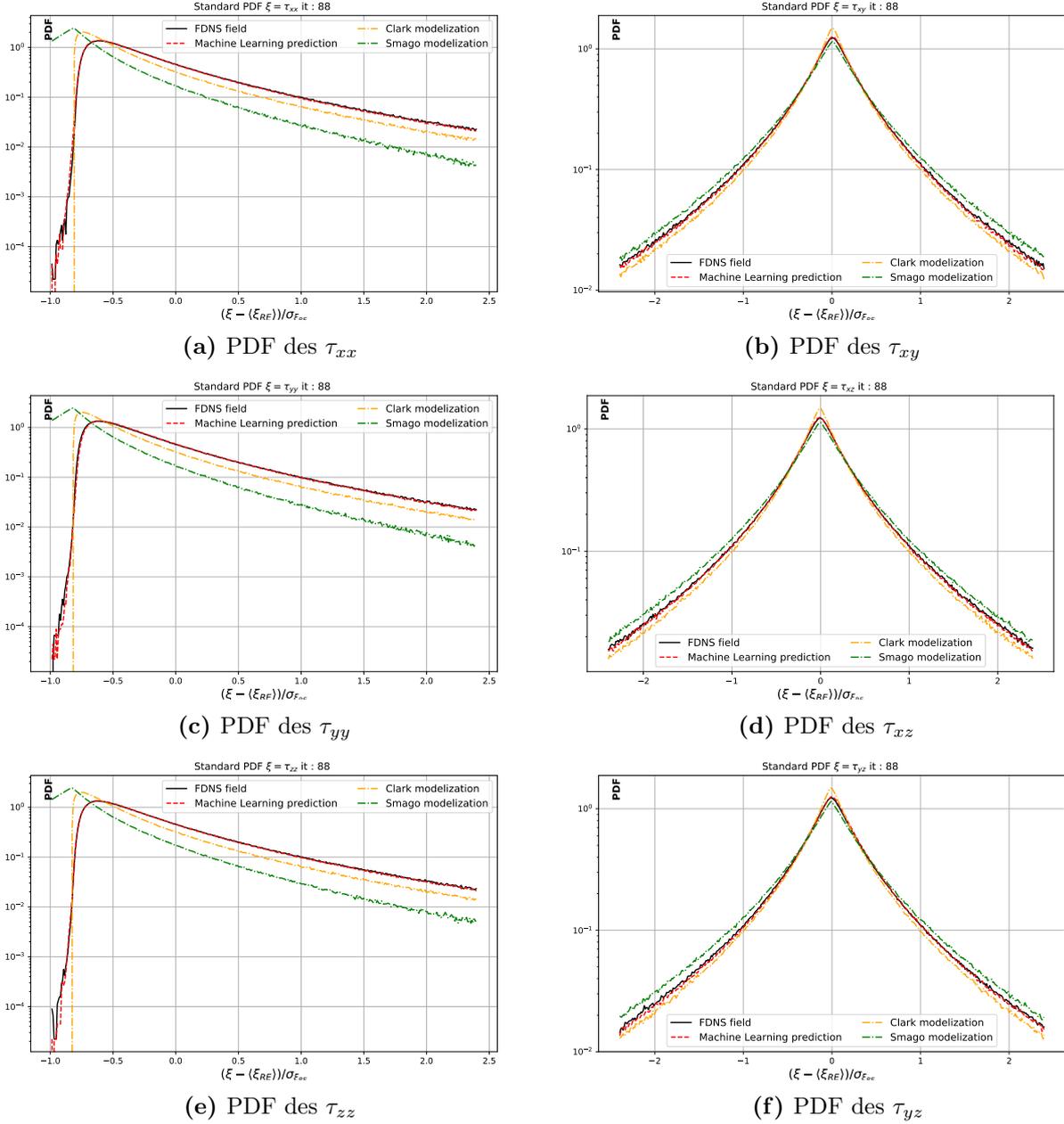


Figure B.3 – À l'itération 88, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

Il est remarquable que les composantes τ_{ij}^{ML} présentent des distributions quasi-identiques aux composantes attendues et que la différence de précision entre composantes diagonales et extra-diagonales est minimale. Les modèles présentent une plus grande difficulté à fournir des tenseurs

dont l'ensemble des distributions sont aussi précises en particulier aux niveaux des composantes diagonales. Ces résultats sont à coupler avec ceux du tableau des corrélations globales au niveau tensoriel Tab.(5.6). Les sorties du modèle de Smagorinsky présentent des corrélations de Pearson de l'ordre de 60 % avec les données FDNS.

PDFs aux niveaux vectoriel et scalaire - Figure Fig.(B.4), nous traçons les PDFs des grandeurs vectorielles (composantes y et z) ainsi que des grandeurs scalaires Π et Ψ calculées par les quatre méthodes. Dans ces figures également, on y voit que le modèle augmenté propose des champs dont les distributions sont plus fidèles aux résultats théoriques que ne le sont les deux modèles algébriques. Ici aussi elles ne tiennent pas compte de la répartition spatiale des erreurs ; il faut donc consulter les corrélations globales tableau Tab.(5.5) en complément de ces distributions.

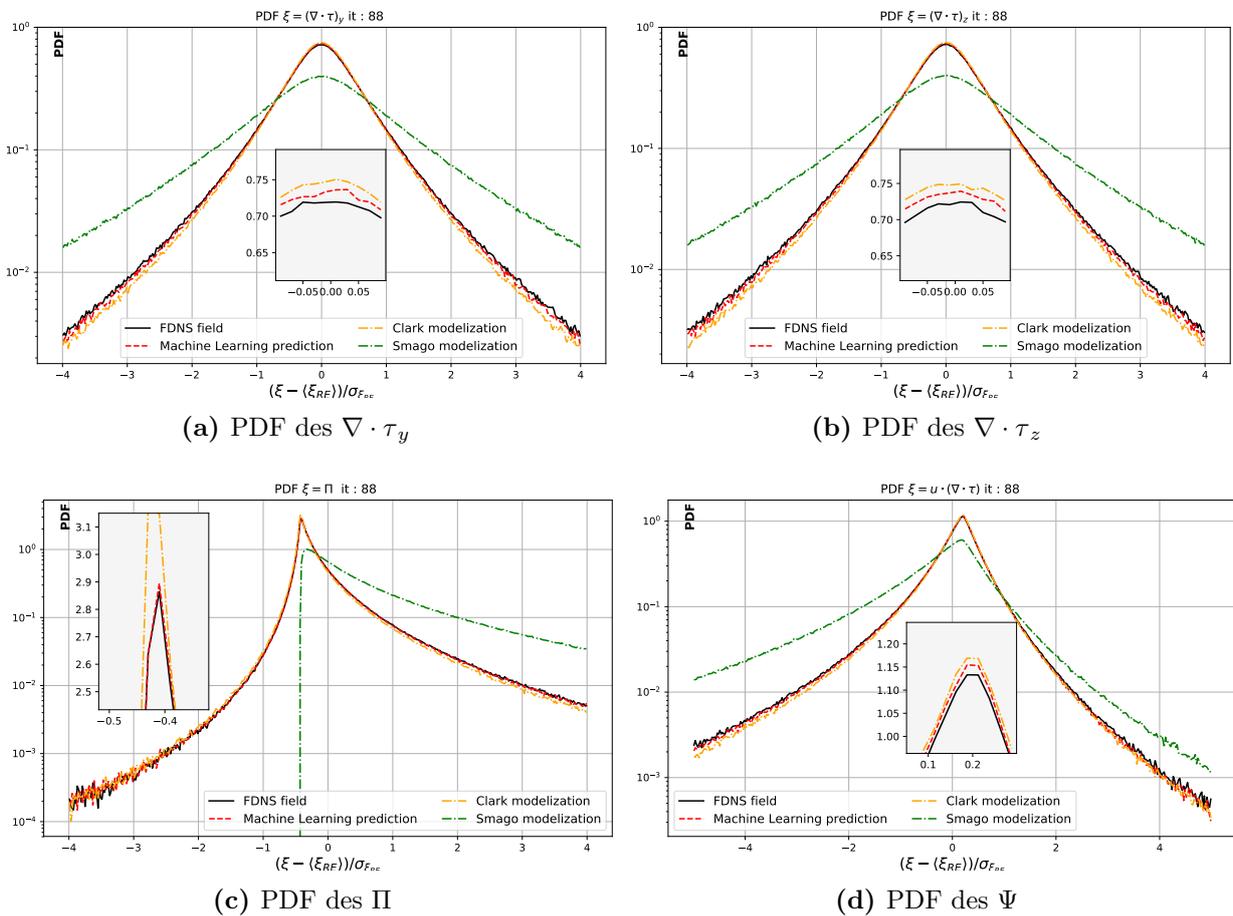


Figure B.4 – À l'itération 88, comparaisons des PDFs des composantes y et z du vecteur $\partial_{x_j} \tau_{ij}$ ainsi que des scalaires Π obtenus par FDNS (courbes continues noires), prédites (ligne brisée rouge) et obtenues à partir des modèles Clark et Smagorinsky (lignes pointillées orange et verte) ; Les PDFs sont standardisées par les moments de la FDNS.

B.1.3 Figures supplémentaires pour l'itération 42

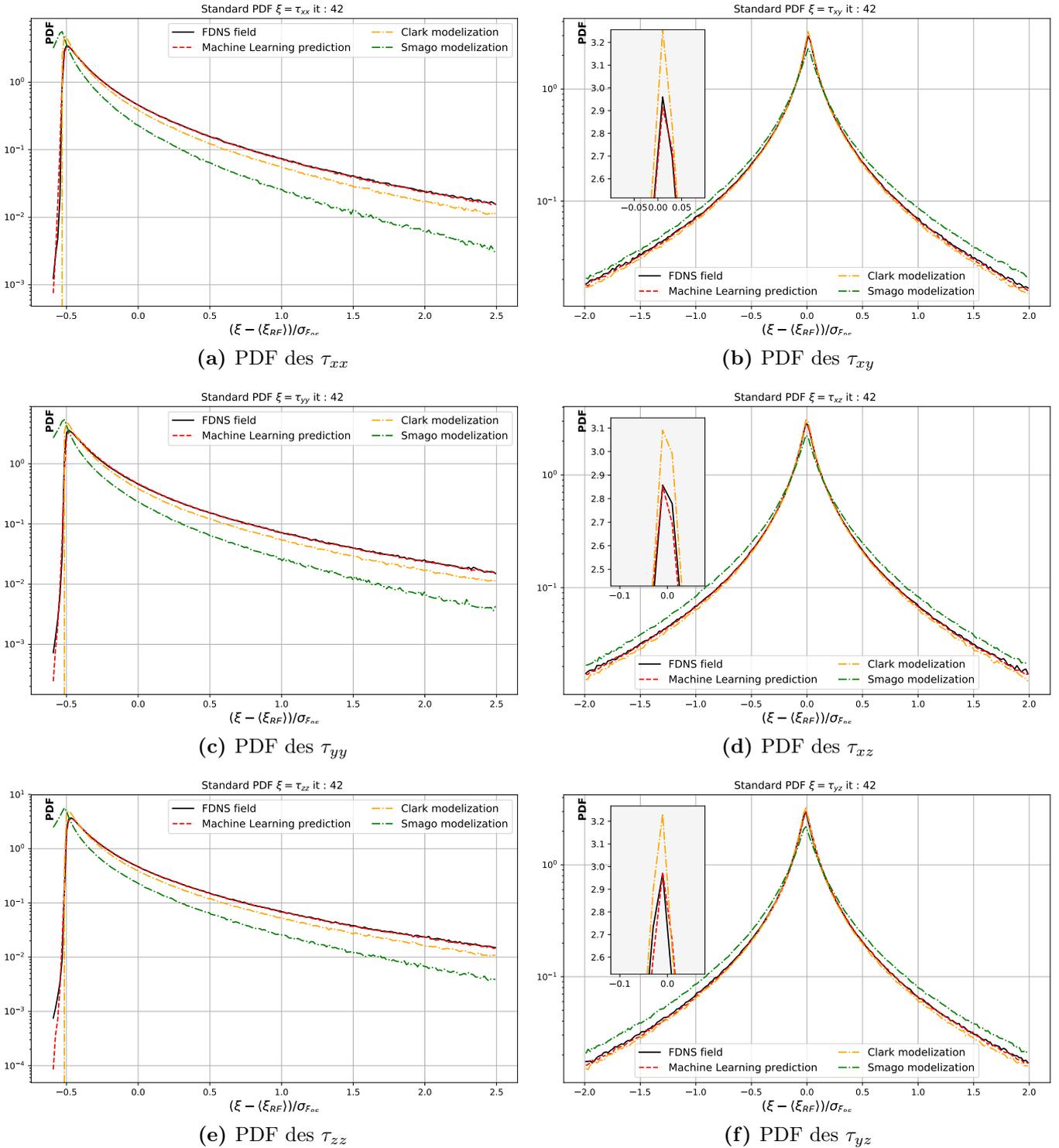


Figure B.5 – À l'itération 42, comparaisons des PDFs des composantes τ_{ij}^{FDNS} (courbe continue noire), avec τ_{ij}^{ML} (ligne brisée rouge), τ_{ij}^{Clark} et τ_{ij}^{Smago} (lignes pointillées orange et verte resp.). Les PDFs sont centrées et réduites par les moyennes et déviations de la FDNS.

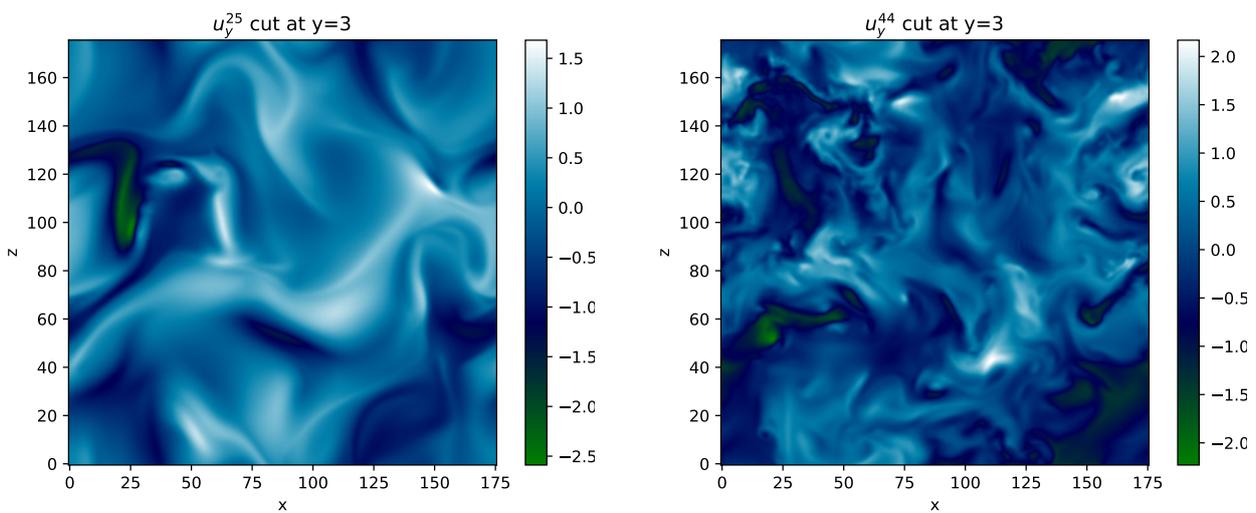
Scores	τ_{xx}	τ_{xy}	τ_{xz}	τ_{yy}	τ_{yz}	τ_{zz}
Pearson	0.99 / 0.99	1.00 / 0.99				
$1 - \mathcal{R}^2$	0.01 / 0.02	0.02 / 0.02	0.02 / 0.02	0.01 / 0.02	0.02 / 0.02	0.01 / 0.02
E_r	0.11 / 0.13	0.15 / 0.13	0.15 / 0.13	0.10 / 0.13	0.14 / 0.13	0.09 / 0.12

Table B.1 – Scores obtenus sur la qualité de la prédiction au niveau tensoriel pour l’itération 25. Les valeurs en noir et **En orange** représentent respectivement les scores sur les prédictions de l’IA et sur la modélisation (Clark) des composantes τ_{ij} .

B.1.4 Figures et scores pour l’itération 25

L’itération 25 est une itération excentrée y compris du point de vue de la turbulence. En effet, comme on peut le voir dans les figures Fig.(B.6a) la turbulence est à l’aube de son développement. Les grandes échelles ne se sont presque pas décomposées. La figure (B.6b) compare le même champ u_y à l’itération 44 dont les scores sont détaillés section 5.1.3 dans lequel les caractéristiques de la turbulence commencent à apparaître. Les contours sont projetés dans le plan $y = 3$.

Les corrélations entre les prédictions basées sur ces champs et les champs théoriques sont ramenés tableau Tab.(B.1). On voit que les scores sont excellents en soit, aussi bien au premier ordre qu’au deuxième ordre. On voit également que les scores basés sur la modélisation de Clark sont très proches ; bien plus que les corrélations aux itérations précédentes. Au delà de



(a) Itération 25 de la simulation de résolution 176

(b) Itération 44 de la simulation de résolution 176

Figure B.6 – Illustrations des champs de vitesse u_y pour l’itération 25 et 44 dans le plan $y = 3$.

la comparaison qui est moins marquée, il est notable que le réseau de neurones est arrivé à prédire l'action des petites échelles aussi rare soient elles. Nous continuons en renseignant les scores pour les niveaux vectoriel et scalaire. Les scores sont présentés dans le tableau Tab.(B.1). On voit clairement que les contractions résonnent avec les erreurs de prédictions contrairement aux modélisations. Nous pensons que les erreurs sur les extrémités des cubes se propagent ; erreurs visibles sur les contours des τ_{xx}^{FDNS} , τ_{xx}^{ML} et τ_{xx}^{Clark} figures Fig.(B.7). Les erreurs sont plus diffuses dans la modélisation de Clark. Une partie des travaux qui suivront cette thèse consistera à tenter d'expliquer ce comportement aux différents niveaux, ainsi que de tenter de les régler en ajoutant des données dans l'apprentissage pour augmenter la richesse de ce dernier et améliorer les prédictions aux tout début du mécanisme de création d'échelles.

B.2 Résultats complémentaires pour la simulation dont Reynolds $Re_\lambda = 240$

B.2.1 Sur l'itération 125

Nous montrons figures Fig.(B.8) les coupes de τ_{yz}^{FDNS} , τ_{yz}^{ML} et τ_{yz}^{Smago} pour illustrer le fait que les modélisations de Smagorinsky sont assez décorrélés des champs théoriques malgré les bonnes distributions

Scores	$\{\nabla \cdot \tau\}_x$	$\{\nabla \cdot \tau\}_y$	$\{\nabla \cdot \tau\}_z$	Π	$u \cdot (\nabla \cdot \tau)$
Pearson	0.88 / 0.96	0.90 / 0.96	0.89 / 0.96	0.97 / 0.98	0.90 / 0.96
$1 - R^2$	0.27 / 0.08	0.23 / 0.08	0.25 / 0.08	0.06 / 0.04	0.22 / 0.07
E_r	0.52 / 0.29	0.48 / 0.28	0.50 / 0.29	0.23 / 0.19	0.46 / 0.27

Table B.2 – Scores obtenus sur la prédiction aux niveaux vectoriel et scalaire pour l'itération 25. Les valeurs en noir et en orange représentent respectivement les scores obtenus par l'IA et par le modèle Clark sur les composantes du vecteur $\partial_{x_j} \tau_{ij}$ et des grandeurs scalaires Π et Ψ .

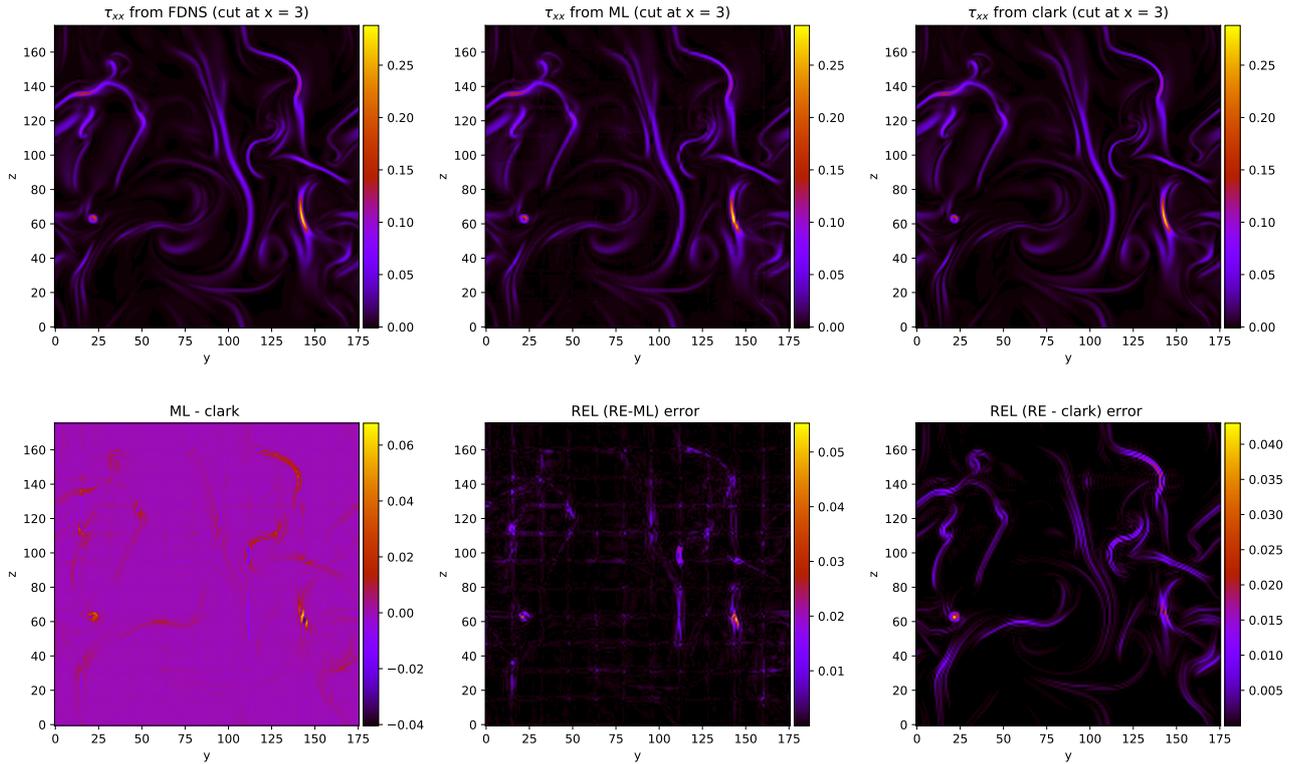


Figure B.7 – Figures du haut : Coupe de la composante τ_{xx} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Clark (à droite). Les échelles de couleurs sont celles de la FDNS. Figures du bas : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Clark et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 0$.

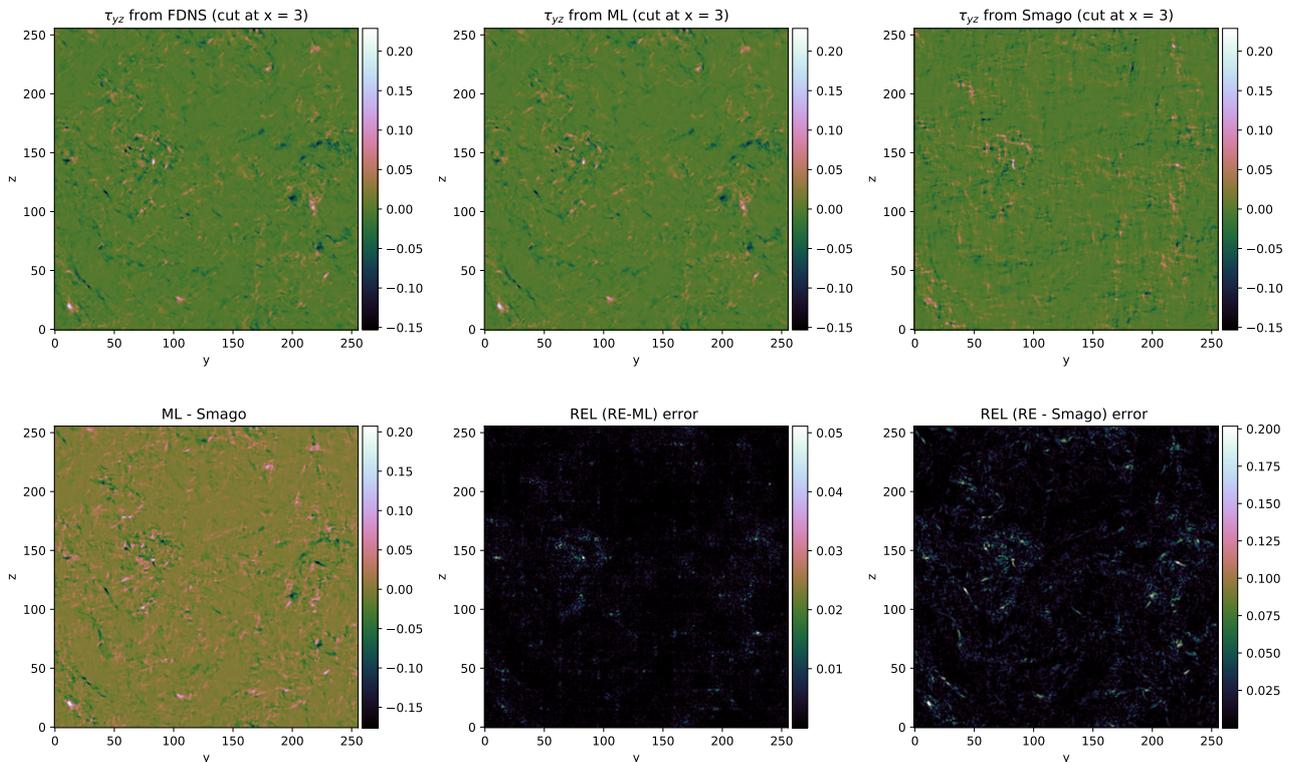


Figure B.8 – Figures du haut : Coupe de la composante τ_{yz} FDNS (à gauche), prédite par IA (au centre), obtenue par le modèle de Smagorinsky (à droite). Les échelles de couleurs sont celles de la FDNS. Figures du bas : écart relatif entre prédiction et modélisation (à gauche), différence en valeur absolue FDNS - ML (au centre) et entre Smagorinsky et FDNS (à droite). Les grandeurs sont visualisées dans le plan $z = 0$.

Scores	L_{xx}	L_{xy}	L_{xz}	L_{yy}	L_{yz}	L_{zz}
Pearson	0.99 / 0.95	0.99 / 0.98	0.99 / 0.98	0.95/ 0.99	0.99 / 0.98	1.00 / 0.95
$1 - \mathcal{R}^2$	0.02 / 0.18	0.01/ 0.04	0.02/ 0.05	0.02/ 0.17	0.02/ 0.04	0.02/ 0.18
E_r	0.16 / 0.42	0.12/ 0.21	0.13/ 0.21	0.15/ 0.41	0.13/ 0.21	0.15/ 0.43

Table B.3 – Scores obtenus sur la qualité du tenseur L_{ij} pour l’itération 61. Les valeurs en noir et **En orange** représentent respectivement les scores sur les prédictions de l’IA et sur la modélisation (Clark).

B.2.2 Sur l’itération 61

Comme précisé dans le corps, nous renseignons les tableaux comparant les scores des corrélations entre L_{ij}^{FDNS} et L_{ij}^{ML} ainsi qu’entre L_{ij}^{FDNS} et L_{ij}^{Clark} . Nous voyons que globalement modélisation et prédictions contiennent l’information relatives aux interactions entres échelles résolues. On note que τ_{ij}^{ML} semble contenir plus d’information sur ce genre d’interaction que τ_{ij}^{Clark} . Les erreurs relatives mesurées sur ce dernier tenseur, sont trois plus fortes au niveau des composantes diagonales ; de même les corrélations au sens de Pearson sont plus faibles à ces diagonales et la variances retrouvées. Ce résultat est la conséquence d’une erreur plus diffuse et plus forte sur les composantes diagonales.

Nous renseignons les mêmes résultats pour la composante C_{ij} cette fois :

Scores	C_{xx}	C_{xy}	C_{xz}	C_{yy}	C_{yz}	C_{zz}
Pearson	0.99 / 0.94	0.99 / 0.98	0.99 / 0.98	0.95/ 0.95	0.99 / 0.98	1.00 / 0.94
$1 - \mathcal{R}^2$	0.02 / 0.16	0.01/ 0.04	0.02/ 0.05	0.02/ 0.15	0.02/ 0.05	0.02/ 0.16
E_r	0.14 / 0.38	0.13/ 0.21	0.13/ 0.22	0.13/ 0.37	0.13/ 0.22	0.14/ 0.38

Table B.4 – Scores obtenus sur la qualité du tenseur C_{ij} pour l’itération 61. Les valeurs en noir et **En orange** représentent respectivement les scores sur les prédictions de l’IA et sur la modélisation (Clark).