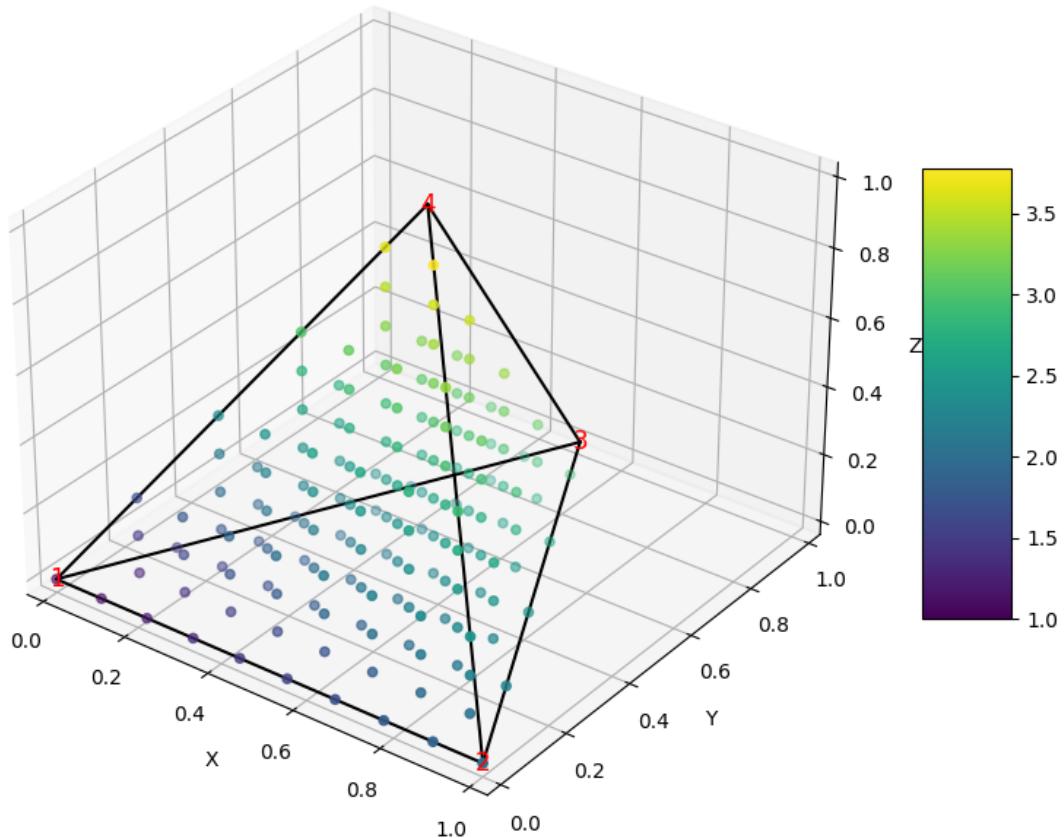


Rapport de stage Aéro 4

Développement de la méthode d'interpolation linéaire et évaluation de ses performances dans l'application de l'aéroacoustique

Stagiaire au sein de l'équipe AAM



Interpolation quadratique dans un tétraèdre

Remerciements

Je tiens à sincèrement remercier mon maître de stage, Carlos MONTILLA, de m'avoir guidé tout au long de ce stage et toujours aidé avec le sourire.

Je tiens également à remercier madame la Présidente du CERFACS, Catherine LAMBERT, pour sa sympathie et pour m'avoir permis d'effectuer ce stage.

Merci à mes amis montagnards et passionnés de mathématiques, Benjamin CANOVAS-ANDRIEUX et Dimitri LANIER ainsi qu'à mon professeur à l'IPSA, Guillaume COUFFIGNAL, pour leur précieuse aide en mathématiques.

Merci à mon tuteur pédagogique, Nadir MESSAI, pour son aide dans ma recherche de stage et au sein du CERFACS.

Merci à Alexis BOUDIN pour ses explications sur l'interpolation d'ordre élevé. Merci à mes amis et collègues Kélian RENOUX, Luc POTIER, Arthur COLOMBIÉ, Guillaume DAVILLER ainsi qu'à l'administration et au CSG pour l'aide qu'ils m'ont apportée au cours de ce stage.

Table des matières

Bilan Technique	4
Introduction	5
1 Le laboratoire de recherche CERFACS	6
2 Implémentation de la méthode linéaire dans Antares	10
2.1 La librairie Antares	10
2.2 Les différentes méthodes d'interpolation	12
2.2.1 La méthode par voisin le plus proche	13
2.2.2 La méthode pondération inverse à la distance (IDW)	13
2.2.3 D'autres méthodes	16
2.2.4 La méthode linéaire	21
2.3 Implémentation de la méthode linéaire	25
2.3.1 La structure générale du code "TreatmentInterpolation"	25
2.3.2 Le pseudo-algorithme	27
2.3.3 Optimisation du temps de calcul	27
2.4 Tests	30
2.4.1 Tests unitaires sur la méthode linéaire	30
2.4.2 Tests sur des cas industriels	32
2.4.3 Tests sur des cas d'aéroacoustique	33
2.4.4 Tests sur les paramètres de la méthode IDW	34
Conclusion	36
Annexes	38
Bibliographie	43

Bilan technique

Ce bilan technique a pour objectif de synthétiser les écarts et leurs causes entre les missions du stage et ce qui a été réalisé. Il permet aussi de proposer ce qui pourrait être fait par la prochaine personne travaillant sur le même sujet.

Fiche de synthèse		Clément THIBAULT - Aéro 4
Sujet de stage	Objectifs	
Influence de la méthode d'interpolation sur la propagation acoustique FWH	<ul style="list-style-type: none"> - Développer une méthode d'interpolation linéaire HPC dans Antares - Évaluer l'influence de la méthode d'interpolation dans la qualité des résultats de propagation acoustique avec l'analogie FWH - Améliorer les performances HPC de la méthode d'interpolation dans Antares 	
Client principal	Outils utilisés	
<ul style="list-style-type: none"> - CERFACS - Date de mise à jour : 15 octobre 2024 	VSCode, Python, Kraken (supercalculateur du CERFACS), Antares, Paraview, Git	
Études réalisées		
<ul style="list-style-type: none"> - Influence de paramètres de la méthode d'interpolation IDW - Différences de DSP entre la méthode IDW et linéaire - Optimisation de la rapidité du traitement 		
Résultats	Explications des écarts possibles entre les résultats et le code initial	
<ul style="list-style-type: none"> - Meilleurs paramètres pour l>IDW se situent autour de N=10 et p=10 - Méthode linéaire généralement plus précise que IDW - Rapidité du code d'interpolation augmentée, pour toutes les méthodes (d'un facteur 105 en linéaire, sur le cas test d'aéroacoustique) 	<ul style="list-style-type: none"> - N=10 et p=10 donnent beaucoup d'information et d'importance aux points proches - La méthode linéaire est d'ordre 1 contrairement à IDW qui n'a pas d'ordre au sens usuel du terme - Pour la rapidité, une amélioration a consisté à ne pas recalculer des coefficients à chaque instant de la solution 	
Difficultés rencontrées	Travaux à poursuivre	
<ul style="list-style-type: none"> - Prise en main des outils - Adapter la méthode linéaire au code déjà existant 	<ul style="list-style-type: none"> - Traiter les maillages 'multi-zones' avec points partagés en linéaire - Implémenter une méthode d'ordre supérieur - Passer le code en parallèle pour améliorer la rapidité 	

Introduction

Ce stage de M1, d'une durée de trois mois, s'est déroulé au CERFACS, un institut reconnu pour son expertise en calcul High Performance Computing (pour calcul haute performance) (HPC). L'objectif principal était d'explorer et d'améliorer les méthodes d'interpolation dans le cadre du post-traitement de simulations numériques, avec un focus particulier sur l'application en aéroacoustique.

Le secteur de la recherche en calcul scientifique est réputé pour son environnement exigeant, mais stimulant, à la base de l'innovation. Le CERFACS, en particulier, est un acteur principal dans le domaine du HPC, collaborant avec de grands industriels et institutions pour développer des solutions à la pointe de la technologie.

J'adore les mathématiques appliquées, la mécanique des fluides et je voulais découvrir le monde de la recherche. Lors d'une présentation des activités au Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS) par nos deux enseignants chercheurs Arthur COLOMBIÉ et Nadir MESSAI, j'ai eu l'occasion de découvrir ce laboratoire et d'y candidater pour mon stage de M1. Carlos MONTILLA, docteur au CERFACS, m'a proposé un sujet sur l'interpolation dans le cas de post-traitement de simulations numériques et son application en aéroacoustique. Le sujet m'a interpellé et c'est avec enthousiasme que j'ai ainsi pu commencer mon stage le 10 juin 2024 au CERFACS.

En quelques mots, le CERFACS est un institut de recherche privé, spécialisé dans le développement de code HPC, financé par sept actionnaires.

Antares[2] est un code d'analyse de données privé sous forme de librairie python, développé au CERFACS en 2012 et dont l'objectif est de réaliser du pré et post-traitement sur des simulations numériques utilisées par le CERFACS, ses actionnaires et autres partenaires. Il contient notamment une fonction d'interpolation, codée en Python.

Mon maître de stage, Carlos MONTILLA, est responsable d'Antares depuis dix mois. Il a notamment fortement contribué au traitement Ffowcs Williams-Hawkings (FWH). La chaîne de calcul aéroacoustique utilise le traitement d'interpolation avant de pouvoir utiliser le traitement FWH (voir 2.4.3).

Mes missions principales lors de ce stage ont été :

- de faire un état des lieux sur les autres méthodes d'interpolation qui seraient implémentables dans Antares (avec les contraintes associées) ;
- d'identifier les meilleurs paramètres pour l'équation Inverse Distance Weighting (IDW), la seule qui était implantée jusqu'alors dans Antares ;
- d'implémenter la méthode linéaire.

Pour bien comprendre l'idée globale, l'objectif est d'interpoler les valeurs aux points d'un maillage 'cible', issu d'une discréttisation de l'espace, en utilisant les valeurs aux points d'un maillage 'source'. Par exemple dans le cadre d'un raffinement de maillage entre deux itérations de calcul ou bien dans le cadre de la création d'une sphère dans un maillage 3D pour l'application des équations de FWH pour la propagation aéroacoustique.

Pour expliquer plus en détail ce stage au CERFACS, je présenterai dans une première partie ce laboratoire de recherche, puis dans une seconde partie j'exposerai le travail que j'ai réalisé sous forme de rapport. Cette seconde partie se décomposera en quatre temps : la présentation de la librairie Antares, les différentes méthodes d'interpolation, l'implémentation de la méthode linéaire dans Antares et finalement les tests de l'ancienne et de la nouvelle méthode.

Chapitre 1

Le laboratoire de recherche CERFACS

Le CERFACS est un laboratoire de recherche privé dont les actionnaires sont Airbus, le Centre National d'Études Spatiales (CNES), Électricité De France (EDF), Météo France, l'Office National d'Études et de Recherche Aérospatiales (ONERA), Safran et TotalEnergies. Il a pour but de développer la simulation numérique par le calcul HPC pour ses actionnaires, mais aussi de faire de la recherche et de former des ingénieurs, chercheurs et doctorants. Il a été créé en 1988 sous le statut de Groupement d'Intérêt Public (GIP), pour devenir une société civile en 1996 et depuis 2021, le CERFACS est une Société par Actions Simplifiées (SAS).

Les deux bâtiments du CERFACS sont situés à la Météopole, dans la partie ouest de Toulouse. Environ 170 personnes y travaillent, dont 20 % de femmes, 50% de doctorants et 20% d'étrangers dont la moitié ne sont pas Européens.

Physiciens, mathématiciens, informaticiens, numériciens et data scientist y travaillent dans cinq équipes :

- parallel s ALGOrithmes & sCientifics sOftware Operational Performances (pour Algorithmes parallèles et logiciels scientifiques performances opérationnelles) (ALGO-COOP)
- Computer Service General (pour Équipe informatique et support utilisateur) (CSG)
- Energy and Safety (pour Énergie et sécurité) (ES)
- Advanced Aerodynamics and Multiphysics (pour Aérodynamique avancée et multiphysique) (AAM)
- modelling climate and GLOBal Change (pour Modélisation du climat et de son changement global) (GLOBE)

La Conception Assistée par Ordinateur (CAO) permet de faire les plans numériques d'un avion par exemple, afin de s'assurer que toutes les pièces fabriquées vont bien s'imbriquer entre elles. Mais cela permet aussi de faire des simulations numériques pour prévoir la tenue structurelle, les forces aérodynamiques, le volume sonore, etc. sans avoir à faire d'onéreux tests.

L'équipe AAM se focalise sur la simulation des écoulements aérodynamiques externes en développant des méthodes numériques avancées et en les appliquant aux avions, fusées, hélicoptères, moteurs, turbomachines, etc. Les liens de l'équipe AAM avec toutes les autres équipes du CERFACS sont forts. Par exemple, les simulations des chambres de combustion faites par l'équipe ES donnent les conditions de sortie dans des moteurs d'avions qui sont ensuite utilisées par l'équipe AAM pour des simulations aéroacoustique. Dans l'autre sens, les simulations aérodynamiques externes des ailes d'avions réalisés par AAM sont ensuite utilisées à GLOBE pour faire des analyses des contrails afin de déterminer leurs impacts sur le climat.

Le CERFACS héberge des supercalculateurs et un serveur sur lequel se trouve l'intranet avec toutes les ressources nécessaires aux employés. Nous y trouvons notamment les liens vers la Qualité de Vie au Travail (QVT), le Comité Social et Économique (CSE), un document "Plan du management de la qualité" et énormément d'autres ressources.

Après une recherche non fructueuse sur l'intranet, j'ai pu demander à la responsable des ressources humaines le document relatant des objectifs Responsabilité Sociétale des Entreprises (RSE). Le document d'actions RSE est segmenté en trois parties principales :

Démarche RSE

"Le Cerfacs n'a pas mis en place une démarche spécifique RSE mais des actions ont été réalisées dans ce cadre, vis-à-vis de l'écoresponsabilité, la qualité de vie au travail et une charte éthique a été incluse dans le règlement intérieur depuis décembre 2023 (intégrant notamment la prévention de la corruption et la gestion des conflits d'intérêts).

En outre, le Cerfacs achète ses fournitures auprès de créateurs solidaires (exemple : Antilope qui compte 32 salariés sur les 49 avec une reconnaissance de « travailleur handicapé »).

Le Cerfacs met à jour au moins deux fois par an et aussi souvent que nécessaire un document unique d'évaluation des risques Cerfacs (avec un plan d'actions de prévention).

Le Cerfacs a nommé des référents : un référent « Santé et Sécurité », un référent « Règlement Général sur la Protection des Données (RGPD) », un référent « Handicap », deux référents « Harcèlement, agissements sexistes et violences ».

Un bilan HSE et RSE est présenté chaque année aux Associés du Cerfacs lors de l'Assemblée d'avril."

Écoresponsabilité

J'ai constaté beaucoup d'actions dans le sens de l'écoresponsabilité lors de mon stage. La plupart sont mentionnées dans le document d'actions RSE.

Un Groupe de Travail « Bilan carbone du CERFACS », composé de 10 volontaires, a été créé. Des bilans carbone ont été réalisés pour les années 2019 et 2021, en suivant la méthodologie proposée par le collectif de laboratoires Labos1point5¹. Depuis mi-2023, le CERFACS collabore avec le Cabinet Take[air] pour établir un plan d'actions visant à réduire son empreinte carbone. Cette collaboration a permis de réaliser le bilan carbone de 2022 et de commencer l'élaboration d'un plan d'action, qui devrait être finalisé à l'automne 2024, après la tenue de plusieurs ateliers. Finalement nous trouvons dans le document un tableau A3 récapitulant les principaux postes d'émission identifiés dans une première colonne puis les objectifs et actions associées dans une seconde colonne.

En parallèle, j'ai entendu parler de cette démarche en discutant avec des thésards. Et j'ai aussi pu trouver la page depuis l'intranet. Elle est divisée en cinq sections : "présentation", "actualités", "émissions", "bilan" et "actions". J'aime beaucoup l'image ci-dessous (figure 1.1), présente dans le document et le site internet, qui montre que le CERFACS réduit ses émissions d'année en année. Cela est probablement dû aux efforts du groupe « Bilan carbone du CERFACS ». L'année 2021 est peu représentative, car grandement marquée par une diminution des missions à cause de la COVID. Mais 2022 nous montre bien que le CERFACS réduit les émissions dues à

1. <https://labos1point5.org>

ses missions, à la fabrication et utilisation de calculateurs externes, aux trajets domicile-travail et au matériel informatique, sujets cibles de la sensibilisation du groupe.

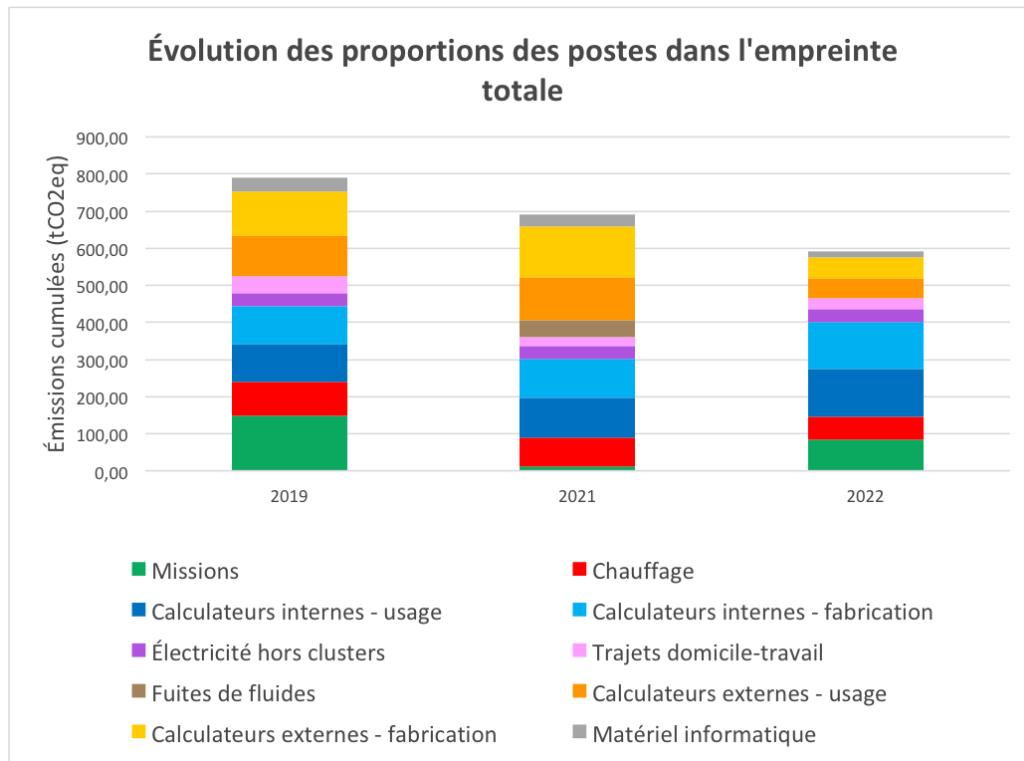


FIGURE 1.1 – Évolution de l'empreinte carbone du CERFACS entre 2019 et 2022

Dans les couloirs du CERFACS, on trouve des posters très clairs sur l'empreinte carbone d'un trajet pour une conférence en fonction des moyens de transport. Les employés sont incités à venir à vélo. Le CERFACS donne une prime pour chaque kilomètre fait à vélo sur le trajet habitation-travail-et vis-versa. Il y a aussi un atelier de réparation et des emplacements sécurisés pour les vélos. La politique de communication n'est pas spécialement tournée vers les efforts pour le climat ou la RSE, mais le CERFACS agit efficacement en ce sens.

Responsabilité sociale / Qualité de vie au travail

"La démarche Qualité de Vie au Travail (QVT) a été lancée au Cerfacs en janvier 2022, avec la mise en place d'un Groupe de Travail : elle a permis d'identifier 6 axes de travail pour améliorer le fonctionnement de la structure et du ressenti des personnes :

- Optimiser l'organisation et gestion du travail au niveau global
- Optimiser l'organisation et gestion du travail au niveau de l'équipe
- Optimiser l'organisation et gestion du travail au niveau personnel
- Assurer un meilleur accueil et support des non permanents
- Favoriser la vie sociale
- Améliorer le confort matériel"

De nouveau, un tableau annexe A4 détaille des exemples d'actions que j'ai pu experimenter (salle de repos, machine à café, endroit pour attacher son vélo, etc).

Un dernier plan d'action a été mis en place sur l'égalité professionnelle entre les femmes et les hommes. Il traite de diverse sujets : l'embauche, la rémunération effective et l'articulation entre l'activité professionnelle et l'exercice de la responsabilité familiale.

Je tiens à mentionner également que j'ai reçu un mail qui lançait un appel au volontariat

pour composer le Comité de Pilotage pour la Prévention des Violences Sexistes et Sexuelles au Travail suite à une réunion d'information réalisée en collaboration avec la médecine du travail.

Mon ressenti personnel sur le cadre de travail est très positif. Être assis dans un bureau climatisé avec une vue sur la campagne participe évidemment à cela. Un des seuls risques à ce type de travail est une mauvaise position qui peut entraîner des problèmes de dos, épaules, etc. Pour pallier cela, il y a des affiches dans chaque bureau indiquant la meilleure position à avoir et proposant des exercices. Ces mêmes informations apparaissent dans le 'livret d'accueil stagiaire' qui m'a été remis le premier jour. Une cantine présente sur le site à quelques minutes à pied, permet de déjeuner et échanger avec les collègues du CERFACS hors cadre professionnel.

Chapitre 2

Implémentation de la méthode linéaire dans Antares

//////// [1]

Le stage a débuté au début du mois de juin 2024. Un ordinateur (mac équipé d'un processeur 2,3 GHz Intel Core i9 8 cœurs) et un PIN-pad générateur d'One-Time-Password (OTP) ont été mis à ma disposition. Des identifiants pour se connecter à un supercalculateur du CERFACS m'ont été donnés. Carlos, mon maître de stage, m'a fourni les informations nécessaires pour comprendre et utiliser Antares [2], une librairie de développement, à l'aide de la documentation en ligne¹. Ensuite j'ai résolu un bug mineur sur Antares, ce qui m'a permis de me familiariser avec Nitrox, le Gitlab hébergé sur le serveur du CERFACS où sont situés Antares et d'autres codes du CERFACS.

Lors de mes essais sur le supercalculateur, j'ai utilisé une partition qui offre une puissance de calcul totale de 723 Tflop/s. Elle est composée de 185 noeuds Intel 2x18 cœurs Skylake, cadencés à 2,3 GHz avec 96 GO de mémoire, et de 40 noeuds Intel 2x38 cœurs IceLake, cadencés à 2,4 GHz avec 256 GO de mémoire. Ces ressources m'ont permis d'effectuer les tests les plus coûteux.

2.1 La librairie Antares

Antares est une librairie Python/C++ développée depuis plus de dix ans au CERFACS. Elle permet de traiter des données de simulation numérique issues de différents solveurs utilisés par le CERFACS et ses partenaires. Ces données sont organisées par Antares de manière unique et commune, ce qui est une grande force, notamment pour le partage et l'optimisation des post-traitements. La bibliothèque inclut une cinquantaine de traitements, tels que la création de vues en coupe et l'interpolation. Elle repose fortement sur la bibliothèque numpy pour tirer parti de ses algorithmes optimisés.

Chaque traitement est une classe composée d'une liste de mots-clefs qui correspondent à chaque paramètre. Certains sont nécessairement à donner par l'utilisateur, d'autres sont initialisés par défaut et peuvent être modifiés ou non. Nous retrouvons les deux paramètres 'source' et 'target' nécessaires au traitement d'interpolation ligne 3 et 4 de l'exemple de code 2.1. Les autres arguments du traitement d'interpolation sont listés plus bas (Tableau 2.1).

Une solution CFD est interprétée par Antares comme une "Base". Cette dernière peut être constituée d'une ou plusieurs zones. Une zone représente un emplacement physique de la solution (par exemple, pour une chambre de combustion, nous pourrions avoir deux zones, régis avec des équations différentes : une avant les injecteurs et une après). Chaque zone peut avoir un ou

1. Documentation d'Antares : cerfacs.fr/antares

plusieurs instants. Un instant est une "capture" de la solution à un instant t . Cela peut être la solution finale de la simulation par exemple. Plusieurs instants peuvent représenter une simulation dynamique, qui est utile dans le cas de l'aéroacoustique par exemple, où nous cherchons l'évolution temporelle de la pression. Dans le cas où la structure du maillage ne change pas entre deux instants, Antares ne garde qu'une seule copie de la structure du maillage. Seules les variables changent entre les instants. On parle dans ce cas d'"instant partagé". Nous y retrouvons finalement les variables et leurs valeurs sur le maillage de la zone et instant en question.

La structure des solutions CFD interprétées dans Antares est illustrée ci-dessous (figure 2.1) :

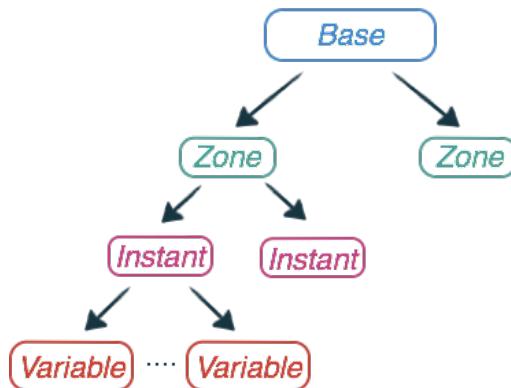


FIGURE 2.1 – Structure des données (source : Antares tutorial)

Voici un exemple d'utilisation d'Antares :

```

1 import antares
2 myt = antares.Treatment('interpolation')
3 myt['source'] = source_base
4 myt['target'] = target_base
5 result = myt.execute()
  
```

Code 2.1 – Exemple simple d'utilisation d'Antares pour interpoler

Dans le cas ci-dessus, `result[0]` correspond à la première zone, `result[0][0]` à son premier instant et `result[0][0][0]` est un tableau numpy manipulable par l'utilisateur correspondant à la première variable du premier instant de la première zone.

Un point important pour l'interpolation linéaire est la connectivité (voir ref 2.2). Si le maillage est non structuré, alors, chaque cellule est définie par un type de forme et par ses sommets. Par exemple, elle peut être donnée par `base[0][0].connectivity['tri']` si la base contient des triangles (équivalent à avoir une connectivité "tri") dans sa zone 0 à l'instant 0.

2.2 Les différentes méthodes d'interpolation

L'interpolation consiste à déterminer une fonction à partir d'un nombre fini de valeurs. En voici un exemple (Figure 2.2) en une Dimension (1D). L'axe x représente la position et l'axe y la valeur des points.

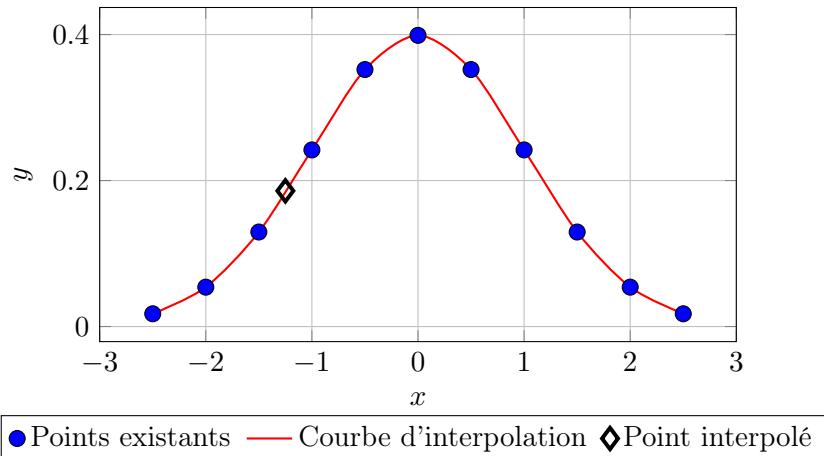


FIGURE 2.2 – Schéma d'interpolation avec des points sur une courbe en cloche

Dans ce paragraphe, nous allons présenter les types d'interpolation [4] implantables dans Antares. Cela implique certaines conditions, notamment sur le type de maillage.

Un maillage structuré est un maillage dont la connectivité entre les points peut être décrite de manière régulière, généralement à l'aide de tableaux multidimensionnels (1D, 2D, 3D). Dans un tel maillage, les points sont organisés en une grille régulière, ce qui permet de naviguer facilement à travers les données sans avoir à spécifier explicitement les relations entre les points.

En revanche, un maillage non-structuré est un maillage où les cellules ne peuvent pas être décrites de manière régulière. Elles sont définies par des formes géométriques quelconques où la connectivité des points ne suit pas un schéma régulier. Cela signifie que chaque cellule doit spécifier explicitement les points qui la composent, dans un ordre spécifique (voir ref A3). Ces maillages sont souvent utilisés pour modéliser des géométries complexes où une grille régulière serait inadaptée.

Dans Antares, nous visons principalement à traiter des maillages non structurés, car ils sont couramment utilisés dans les simulations modernes en raison de leur flexibilité à représenter des géométries complexes. Une autre condition est que la méthode d'interpolation doit être applicable en 1D, 2D et 3D. Un facteur à prendre en compte dans la méthode que l'on souhaite implémenter est aussi le temps de calcul, appelé 'coût'.

La caractérisation mathématique des fonctions à interpoler est le paramètre le plus important à prendre en compte. Cependant, les équations dont sont issues les solutions numériques en entrée dans Antares sont très difficiles à caractériser mathématiquement (tel que l'équation de Naviers-Stokes) et le niveau en mathématiques est trop élevé pour pouvoir se plonger sur ce problème [15]. C'est pour cela qu'il n'y aura pas beaucoup de résultats mathématiques à présenter dans cette partie. Ces méthodes ont déjà été implémentées et testées pour d'autres codes de simulation numérique, ce qui permet de nous donner une bonne idée des résultats que nous pouvons espérer.

Un logiciel créé en 1997 au CERFACS, nommé "hip" permet de convertir des maillages multi-blocs structurés en maillages non structurés. Différentes méthodes d'interpolation y sont codées et décrites dans le guide utilisateur [24]. Cette référence pourrait être lue avant d'implémenter une potentielle future interpolation dans Antares.

La plupart des méthodes d'interpolation peuvent s'écrire sous la forme

$$\hat{f}(x) = \sum_{i=1}^M w_i(x) \cdot f(x_i) \quad (2.1)$$

où :

- $\hat{f}(x)$ est la valeur interpolée à la position x ,
- $f(x_i)$ est la valeur connue aux points de données x_i ,
- $w_i(x)$ est le poids associé à $f(x_i)$,
- M est le nombre total de points de données utilisés pour l'interpolation.

2.2.1 La méthode par voisin le plus proche

Cette première méthode est simple : nous prenons comme valeur v d'interpolation au point p la valeur v du point le plus proche de p . En voici quelques illustrations :

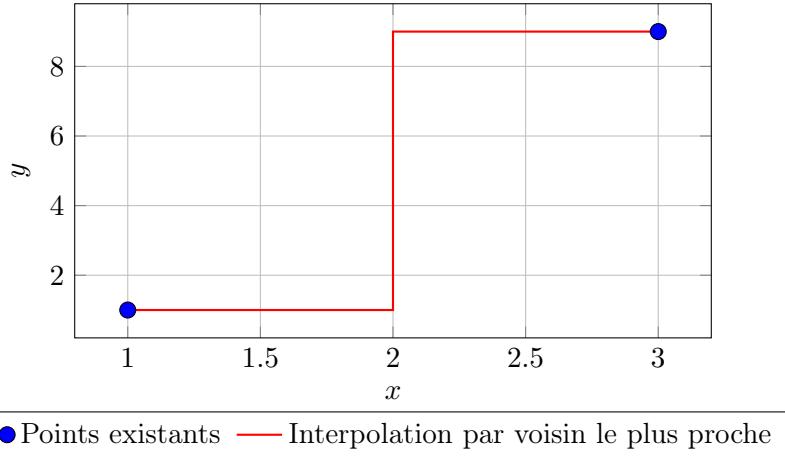


FIGURE 2.3 – Schéma d'interpolation par voisin le plus proche

La courbe rouge représente l'interpolation sur tout le domaine x entre les deux points. Cela représente les coordonnées (x, y) de tous les potentiels points d'interpolation. Cette méthode est discontinue et peu précise pour la plupart des fonctions.

2.2.2 La méthode pondération inverse à la distance (IDW)

Probablement l'interpolation la plus simple après la méthode du voisin le plus proche (toujours dans notre cas d'application), cette méthode est la seule qui était implémentée dans Antares (elle inclut aussi la méthode du voisin le plus proche comme expliqué plus tard dans cette section). Elle a pour formule :

$$\hat{f}(x) = \begin{cases} \frac{\sum_{i=1}^M \frac{f(x_i)}{d(x, x_i)^p}}{\sum_{i=1}^M \frac{1}{d(x, x_i)^p}}, & \text{si } d(x, x_i) \neq 0 \text{ pour tout } i, \\ f(x_j), & \text{si } d(x, x_j) = 0 \text{ pour certains } j. \end{cases} \quad (2.2)$$

$$\text{avec } w_i(\mathbf{x}) = \frac{1}{d(\mathbf{x}, \mathbf{x}_i)^p}$$

où :

- $\hat{f}(x)$ est la valeur interpolée à la position x ,
- $f(x_i)$ est la valeur connue aux points de données x_i ,
- $d(x, x_i)$ est la distance entre x et x_i ,
- p est le paramètre de puissance,
- M est le nombre total de points de données.

En voici des illustrations 1D :

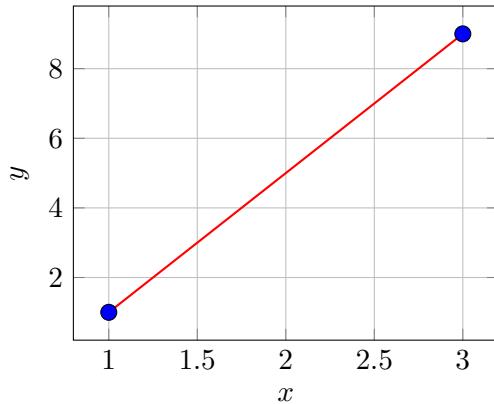


FIGURE 2.4 – interpolation IDW M=2, p=1

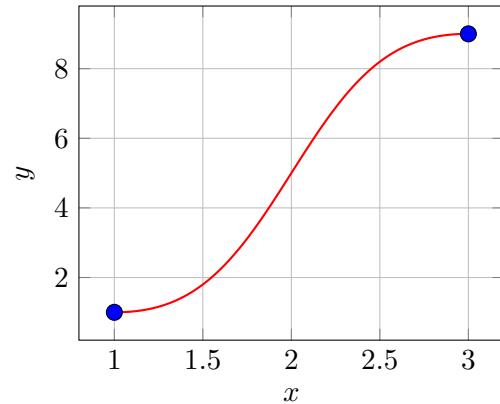


FIGURE 2.5 – interpolation IDW M=2, p=2

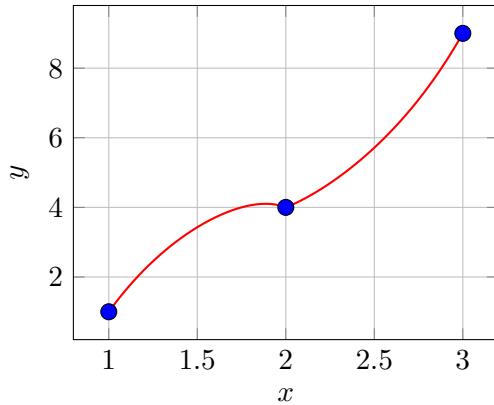


FIGURE 2.6 – interpolation IDW M=3, p=1

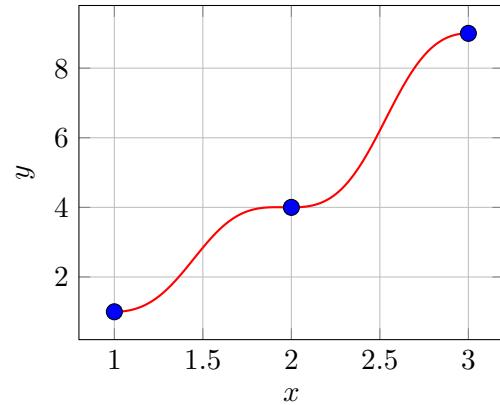


FIGURE 2.7 – interpolation IDW M=3, p=2



Dans notre cas, nous ne prenons pas en compte tous les M points du maillage source. Nous nous limitons à un nombre N de points les plus proches. Nous remarquons que pour $N = 1$, nous retrouvons la méthode du voisin le plus proche, pour tout p . Et pour $N = 2$ et $p = 1$, toujours en 1D, nous retrouvons la méthode linéaire. Limiter le nombre de points dans le calcul implique une discontinuité lorsque nous changeons de point. Pour illustrer cela, voici une figure 1D (figure 2.8) où tous les points sont espacés d'une unité (seuls trois points sont représentés). Avec $M=\infty$ et $N=3$, cela revient ici à changer l'un des trois points de calcul pour une valeur de x égale à 0,5 modulo un. Le modulo un est représenté par une ligne en pointillés noirs.

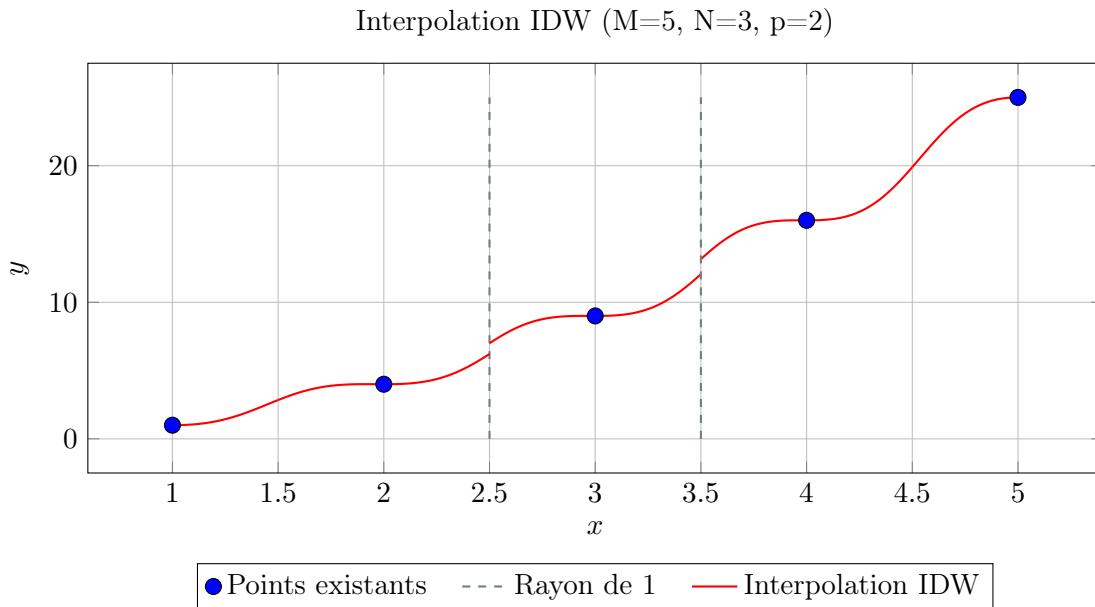


FIGURE 2.8 – Schéma d’interpolation IDW avec $N=3$

Nous observons une discontinuité en 2,5 et 3,5. La discontinuité peut être un grand problème pour la stabilité des schémas numériques.

Nous pouvons aussi imaginer que 2 points, A et B, à droite et à gauche d’une ligne verticale de 3 points (les plus proches), prendront la même valeur pour $N = 3$:

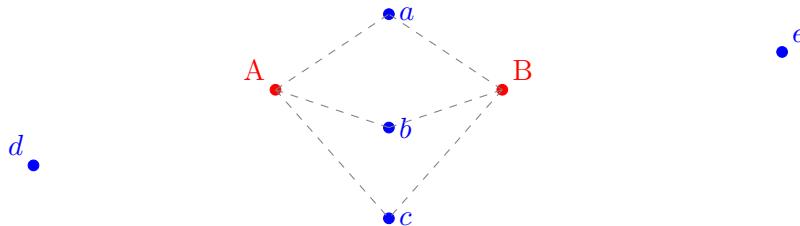


FIGURE 2.9 – Schéma montrant la non-directivité d’IDW

Ce qui n’est pas cohérent dans le cas où le gradient selon l’horizontal n’est pas nul. Un des problèmes avec la méthode IDW est qu’« un biais directionnel dans les solutions de la base de données peut décaler les résultats interpolés » [25] [22]. Cependant, nous pourrions imaginer des solutions pour certains cas, comme, en 2D, prendre, autour du point cible, trois points source répartis dans plus de 180° . [32] Lorsque toutes les informations selon un axe de l’espace ne se situent que d’un côté, le gradient ne sera pas pris en compte et l’erreur sera élevée. Par exemple, cela est généralement le cas lorsque le maillage cible a des points qui ne sont pas compris dans le maillage source.

D’autres méthodes dérivées ou similaires existent pour évaluer nos poids (les poids sont les distances inverses à la puissance p dans notre cas). La plus intéressante serait celle dite de Franke-Little. Elle consiste à utiliser une distance maximale autour du point au-delà de laquelle les autres points ne sont pas pris en compte. Autrement dit, on utilise un cercle (dans le cas 2D) d’un certain rayon pour déterminer quels points nous sont utiles pour l’interpolation. Dans ce cas, le nombre de points est variable. J’ai considéré subjectivement que cette méthode n’était pas intéressante. La méthode confrontée à un maillage ayant une différence de raffinement intrinsèque importante ne prendrait aucun point à certains endroits et dans d’autres lieux, un trop grand nombre de points devrait être calculé.

2.2.3 D'autres méthodes

a. Interpolation de Lagrange [5, 18] *Type : Polynomiale*

L'interpolation de Lagrange utilise un polynôme unique qui passe par n points donnés. Elle peut devenir imprécise pour un grand nombre de points à cause de l'effet Runge. Sa généralisation en 2D et 3D la rend coûteuse. La formule de Lagrange est donnée par :

$$L(x) = \sum_{i=0}^n y_i \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \quad (2.3)$$

b. Interpolation d'Hermite [3] *Type : Polynomiale*

L'interpolation d'Hermite est une extension de l'interpolation de Lagrange, prenant en compte non seulement les valeurs des points, mais aussi leurs dérivées. Pour cela, il faut reprendre le polynôme de Lagrange et imposer la continuité des dérivées. Cela permet de créer un polynôme d'interpolation plus lisse. Cependant, elle est coûteuse, surtout lorsqu'elle est étendue aux ordres supérieurs. La formule de l'interpolation d'Hermite est donnée par :

$$H(x) = \sum_{i=0}^n [y_i h_i(x) + y'_i h'_i(x)] \quad (2.4)$$

où $h_i(x)$ et $h'_i(x)$ sont des fonctions de base Hermite.

c. Interpolation quadratique *Type : Polynomiale*

L'interpolation quadratique est une méthode qui utilise des polynômes de degré deux pour calculer des valeurs intermédiaires entre les points de données. La formule 1D de l'interpolation quadratique pour un ensemble de trois points $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ est donnée par :

$$f(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)^2 \quad (2.5)$$

où les coefficients a_0, a_1, a_2 sont déterminés par les valeurs en x_0, x_1, x_2 .

Dans le cas des maillages non-structurés en 2D ou 3D, l'interpolation quadratique peut être étendue à une forme plus générale. Par exemple, pour un maillage non-structuré en 2D, la formule peut être écrite comme suit :

$$f(x, y) = a_0 + a_1(x - x_1) + a_2(y - y_1) + a_3(x - x_1)^2 + a_4(y - y_1)^2 + a_5(x - x_1)(y - y_1) \quad (2.6)$$

d. Interpolation cubique [31]

L'interpolation cubique est une méthode qui utilise des polynômes de degré trois pour calculer des valeurs intermédiaires entre les points de données. Cette méthode est souvent un bon compromis entre complexité et précision et pourrait être un bon candidat pour la prochaine méthode à implémenter. L'interpolation cubique peut être réalisée en utilisant soit des polynômes de Lagrange, soit des splines cubiques, soit un algorithme de convolution cubique. La formule 1D de l'interpolation cubique entre quatre points x_0, x_1, x_2, x_3 est donnée par :

$$f(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)^2 + a_3(x - x_1)^3 \quad (2.7)$$

où les coefficients a_0, a_1, a_2, a_3 sont déterminés par les valeurs et les dérivées en x_0, x_1, x_2, x_3 .

e. Splines cubiques [16] *Type : Polynomiale*

Les splines cubiques sont une méthode d'interpolation qui utilise des polynômes de degré trois, ajustés de manière à garantir la continuité des premières et deuxièmes dérivées. Cela assure une transition lisse entre les segments de la courbe. Les splines sont stables et précises, notamment lorsqu'il est nécessaire de respecter certaines conditions aux frontières.

La formule 1D pour une spline cubique entre deux points x_i et x_{i+1} est donnée par :

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \quad (2.8)$$

où les coefficients a_i, b_i, c_i, d_i sont déterminés de manière à ce que les fonctions soient continues, ainsi que leurs premières et deuxièmes dérivées, aux points de jonction.

Dans le cas de plusieurs segments, les splines cubiques forment une courbe composée de ces polynômes, ajustée de manière à minimiser la courbure totale tout en passant par tous les points de données.

f. NURBS [27, 7] *Type : Polynomiale*

Les Non Uniform Rational Basis Splines (pour B-splines rationnelles non uniformes) (NURBS) pourrait être un bon candidat pour l'interpolation. C'est une généralisation des splines, permettant de représenter des formes géométriques simples comme complexes. Elles sont notamment utilisées en CAO et pour l'interpolation de surfaces. La formule 1D pour une courbe NURBS est :

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \cdot w_i \cdot P_i}{\sum_{i=0}^n N_{i,p}(u) \cdot w_i} \quad (2.9)$$

où :

- $C(u)$: est la courbe NURBS paramétrée par u .
- $N_{i,p}(u)$: est la base de la spline B-spline de degré p associée au nœud i .
- w_i : est le poids associé au point de contrôle P_i .
- P_i : est le point de contrôle correspondant.

Définissons deux termes utiles pour caractériser les méthodes suivantes : - Autocorrélation spatiale : Mesure du degré auquel les données sont agglomérées ou désordonnées dans l'espace. Voici une application d'autocorrelation (filtre gaussien) sur des données :

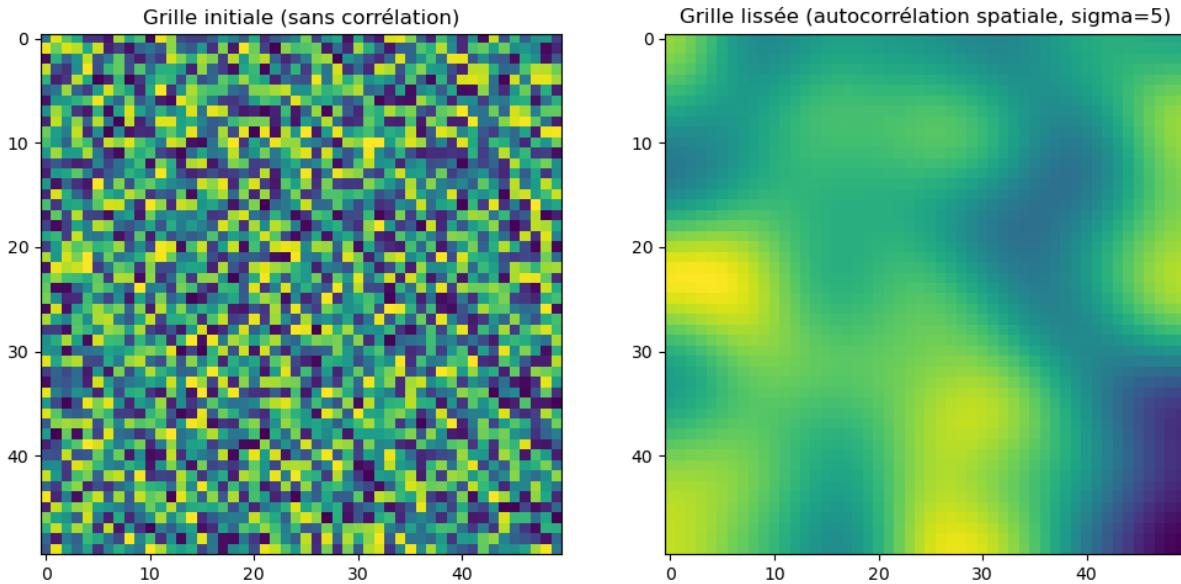


FIGURE 2.10 – Illustration d’application de l’autocorrélation spatiale

- Géostatistique : C'est une méthode d'analyse statistique utilisée pour modéliser et prédire des phénomènes spatiaux. Dans le cadre de l'interpolation, la géostatistique prend en compte l'autocorrélation spatiale des données pour effectuer des prédictions plus précises dans des régions non mesurées, en utilisant des techniques comme le krigage.

g. Kriging et Cokriging [25, 30] *Type : Géostatistique*

Le kriging (ou krigeage en français) est une méthode d'interpolation géostatistique qui génère une estimation statistique des variables spatiales en tenant compte de l'autocorrélation spatiale. Cette méthode est particulièrement efficace pour capturer les comportements non linéaires des variables dans un ensemble de données.

La formule de base pour le kriging est :

$$Z(x) = \sum_{i=1}^N \lambda_i \cdot Z(x_i) \quad (2.10)$$

où :

- $Z(x)$: est la valeur interpolée au point x . - λ_i : sont les poids associés aux points de données x_i , calculés en fonction de la covariance spatiale. - $Z(x_i)$: est la valeur mesurée au point x_i .

Le kriging permet non seulement de prédire les valeurs inconnues, mais aussi d'estimer l'incertitude associée à ces prédictions, ce qui en fait un outil puissant pour l'interpolation dans des domaines où les relations spatiales sont complexes.

Le cokriging est une extension multivariée du kriging, qui prend en compte la corrélation spatiale entre plusieurs variables. Il permet de modéliser simultanément plusieurs variables en utilisant des informations supplémentaires sur leurs corrélations spatiales. Cela peut améliorer la précision de l'interpolation dans les cas où plusieurs variables sont disponibles et corrélées.

La formule pour le cokriging est similaire à celle du kriging, mais elle inclut des termes pour chaque variable considérée, ainsi que leurs covariances croisées.

Cette méthode est particulièrement recommandée pour les applications où les variables ont des relations spatiales complexes et non linéaires, comme dans les simulations CFD (Computational Fluid Dynamics).

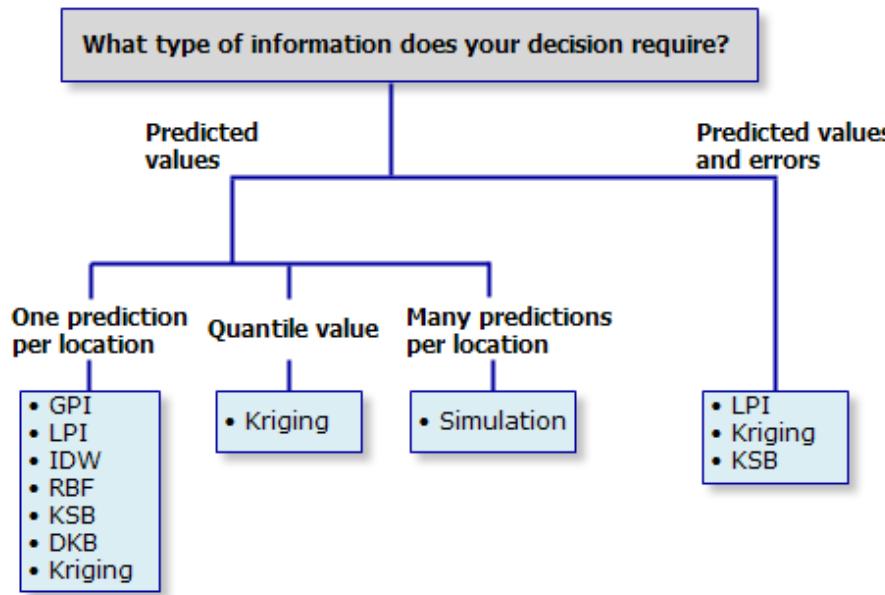


FIGURE 2.11 – Arbre des méthodes d'interpolation géostatistique (source : pro.arcgis.com)

a. Kriging et Cokriging Le kriging (ou krigeage en français) est une méthode d'interpolation géostatistique qui génère une estimation statistique des variables spatiales. Le cokriging est son extension multivariée.

La méthode kriging pourrait être un excellent candidat. Voici une traduction d'un extrait d'article discutant autour des résultats de cette méthode : "La méthode linéaire est simple, robuste et constitue un choix d'interpolation acceptable pour les premières estimations approximatives ou si les variables de la solution de la base de données dépendent linéairement des variables de l'espace de conception. La méthode IDW ne présente aucun avantage par rapport à la méthode linéaire plus simple et n'est pas recommandée pour l'interpolation des solutions CFD. La méthode de Krigeage donne les résultats interpolés les plus précis et permet de saisir le comportement non linéaire des variables de la solution de la base de données. Pour le cas d'essai AS-202, la différence maximale en pourcentage entre le taux de chauffage interpolé par la méthode de Krigeage et le taux de réchauffement CFD exact est inférieure à l'incertitude du taux de réchauffement CFD lui-même." [25]

Le cokriging pourrait être candidat dans le cas où il y aurait une corrélation spatiale entre plusieurs variables. "Le cokriging est l'extension multivariée du formalisme de Kriging, qui permet de traiter simultanément deux variables ou plus définies sur le même domaine et de prendre en compte des informations supplémentaires sur la corrélation spatiale entre les variables." [30] Voici une illustration du résultat de différentes méthodes d'interpolation venant du même article :

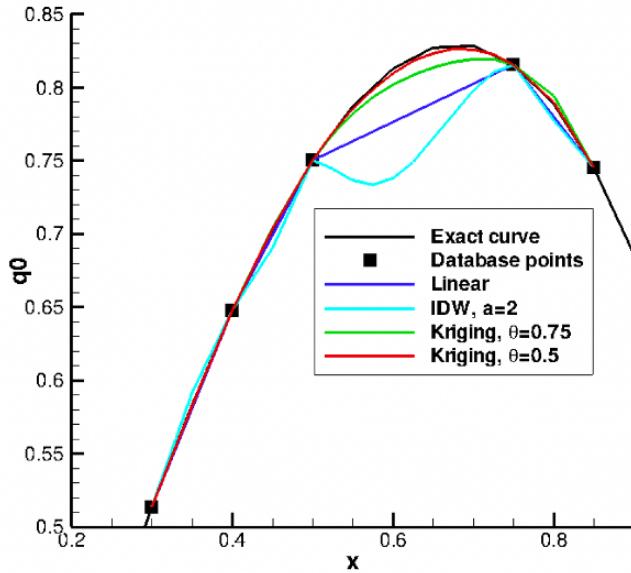


FIGURE 2.12 – Comparaison de différentes méthodes d’interpolation sur des données 1D (source [25])

h. RBF La Radial Basis Function (pour fonction de base radiale) (RBF) est une fonction basée sur les distances telle que la méthode IDW. Cependant, elle peut utiliser différents types de bases radiales pour imaginer la solution, telle que la base gaussienne, multiquadratique, etc. C'est un excellent candidat pour une amélioration de l'interpolation IDW. Il faudrait ensuite trouver le meilleur noyau avec les meilleurs paramètres pour les cas d'applications rencontrés dans Antares. Pour des cas spécifiques tels que l'aéroacoustique, un petit algorithme itératif ou à base d'ajustement par la méthode des moindres carrés (ref 2.2.3) peut être mis en place pour trouver ces paramètres. Voici la formule de la RBF pour un noyau ϕ gaussien :

$$f(\mathbf{x}) = \sum_{i=1}^N w_i \varphi(\|\mathbf{x} - \mathbf{x}_i\|), \varphi(r) = e^{-(\epsilon r)^2} \quad (2.11)$$

Cette interpolation est utilisée et expliquée par un docteur en maths [19]. "We specifically use the Gaussian RBF and the Exponential Dot Product Kernels to do the interpolations, and approximate the benchmark called Franke's Function." [6]

Pour prochaine méthode : Des codes open source d’interpolation d’ordre élevé sont disponibles sur GitHub (en Julia [12] et Python [6] pour RBF et Noyaux de produits de points exponentiels).

Les méthode par moindres carrés Cette méthode minimise l’erreur quadratique pour créer une approximation des données. Elle est appliquée sur une fonction d’approximation paramétrable. Des variantes plus performantes sont décrites dans le guide utilisateur de hip. "Piecewise linear and minimumnorm interpolation approaches are faster than the linear least squares." [24]

Approfondir des recherches sur la méthode Moving Least Squares (pour Moindres carrés mobiles) (MLS) [23] [21] pourrait être utilisé intéressant.

MISCOG Plusieurs méthodes d’interpolation d’ordre élevé assez innovantes telles que les méthodes Multi-Instance Solver Coupled trough Overlapping Grids (pour Solveur multi-instances couplé à travers des grilles superposées) (MISCOG) sont déjà implémentées dans d’autres codes du CERFACS et un article les détails [20]. Elles servent généralement pour d’autres

applications comme pour calculer un flux au travers d'une cellule à l'aide d'une intégration sur une surface. Mais certaines parties sont intéressantes pour notre cas.

2.2.4 La méthode linéaire

"La méthode linéaire est simple, robuste et constitue un choix d'interpolation acceptable pour les premières estimations approximatives ou si les variables de la solution de la base de données dépendent linéairement des variables de l'espace de conception" [25]. La littérature [13] s'accorde à dire que l'interpolation linéaire, est la plus basique après plus proche voisin et IDW. J'entends par basique simple à implémenter, demandant peu de ressources, robuste et moyennement précise pour de l'ordre élevé. C'est pour cela que le CERFACS voulait l'implémenter dans Antares. C'est aussi la plus utilisée par Airbus, Safran et d'autres industriels (via d'autres codes qu'Antares). Cela est donc aussi intéressant d'avoir une interpolation linéaire directement dans Antares.

1D

Linéaire En 1D, l'interpolation linéaire est simple : c'est la moyenne pondérée linéairement par la distance, des valeurs des deux points les plus proches. Supposons que nous voulons interpoler une valeur d'un point p entre deux points a et b dans un espace 1D et que nous représentions leurs valeurs dans une deuxième dimension y . Nous aurons alors pour formule :

$$y_p = \frac{|x_b - x_p|}{|x_b - x_a|} \cdot y_a + \frac{|x_a - x_p|}{|x_b - x_a|} \cdot y_b \quad (2.12)$$

où y_p représente la valeur interpolée à la position x_p , et (x_a, y_a) et (x_b, y_b) sont les points de référence. J'ai écrit cette formule afin qu'elle soit symétrique par rapport aux points a et b , pour qu'ils jouent le même rôle. Ainsi elle s'entendra plus intuitivement dans des dimensions supérieures.

$\frac{|x_b - x_p|}{|x_b - x_a|}$ est le poids pour y_a basé sur la distance relative de x_p à x_b .

$\frac{|x_p - x_a|}{|x_b - x_a|}$ est le poids pour y_b basé sur la distance relative de x_p à x_a .

Ces deux termes sont pondérés de manière à ce que leur somme soit toujours égale à 1, ce qui garantit que l'interpolation est correcte et symétrique par rapport à a et b .

2D En 2D, nous devons nous baser sur des surfaces, extraites de formes pour pouvoir effectuer cette pondération. En CFD, ces formes sont appelées cellules et leurs sommets nœuds. Dans notre cas, nous considérons que les variables du maillage sont contenues au niveau des nœuds. Il existe 3 principaux types de cellules en 2D : les triangles, les rectangles des maillages structurés (dans ce cas, la méthode est dite bilinéaire) et les quadrilatères (non croisés). Nous trouvons de plus en plus de polygones à N arrêtes dans les maillages. Mais par souci de simplicité, l'interpolation linéaire se limite aux trois principaux types de cellules, les polygones pouvant être décomposés en triangles.

Triangle : Barycentrique Pour le triangle, la méthode pour trouver la valeur au point à interpoler p est celle dite du barycentre (barycentrique). Elle est bien documentée.

Visuellement, il faut faire la somme des valeurs aux points pondérés par la surface opposée et pondérer le tout par la surface du triangle.

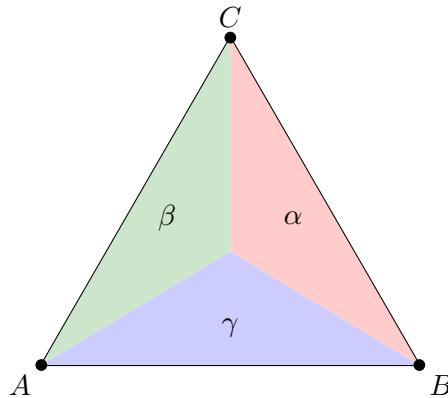


FIGURE 2.13 – Schéma d’interpolation barycentrique

Le calcul est le suivant :

$$\frac{\alpha f(A) + \beta f(B) + \gamma f(C)}{\alpha + \beta + \gamma} \quad (2.13)$$

Bien sûr, les triangles sont quelconques, mais la formule reste la même.

Rectangle : Bilinéaire En ce qui concerne l’interpolation bilinéaire sur un rectangle, nous la trouvons aussi facilement dans la littérature. La formule est l’extension de celle pour les triangles :

$$f(x, y) = \frac{(x_2-x)(y_2-y)f_{11} + (x-x_1)(y_2-y)f_{21} + (x_2-x)(y-y_1)f_{12} + (x-x_1)(y-y_1)f_{22}}{(x_2-x_1)(y_2-y_1)} \quad (2.14)$$

Nous supposons que le point est bien à l’intérieur du rectangle.

Ici les surfaces sont directement calculées dans $f(x, y)$.

Cela correspond à l’addition de deux interpolations linéaires. Nous trouvons souvent dans la littérature une équation analytique où tous les sommets ne jouent pas le même rôle, mais il est plus simple de faire un calcul de poids pour pouvoir ensuite faire une moyenne pondérée. Visuellement, nous créons cette fois des traits parallèles au passant par le point d’interpolation et nous additionnons, de manière pondérée, les 4 surfaces multipliées chacune par leur sommet opposé respectif.

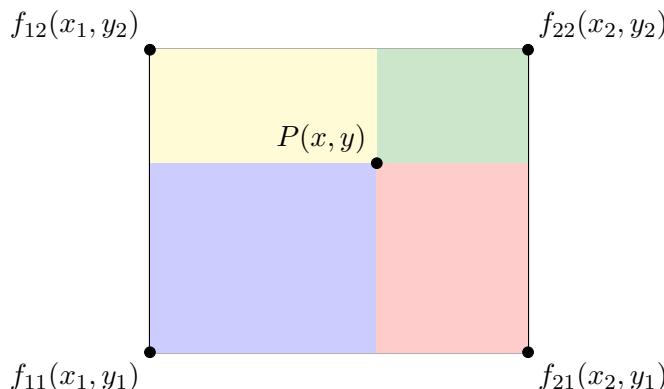


FIGURE 2.14 – Schéma d’interpolation bilinéaire

Quadrilatère Regardons maintenant le dernier type de cellule 2D rencontrée dans les solutions traitées par Antares : les quadrilatères. Pour cela, je n'ai pas trouvé de méthode satisfaisante dans la littérature [26]. Après plusieurs essais sur papier, je me suis concentré sur le fait que la méthode devait être continue, ce qui implique notamment que la valeur du point à interpoler doit tendre vers la valeur d'un sommet lorsque sa distance à ce dernier tend vers 0. Une première vérification de la linéarité est aussi de vérifier qu'un point au milieu d'une cellule 2D a comme valeur la moyenne de ses sommets. Via cette démarche, j'ai imaginé, graphiquement, tracer des traits entre le point à interpoler et les sommets de la cellule dans laquelle il se situe (tel que pour l'interpolation Barycentrique). Cela permet de ne créer que quatre sous formes. Ensuite pour déterminer le poids associé au sommet s_1 , il faut multiplier entre elles les deux surfaces qui lui sont opposées, et bien entendu, le pondérer une fois les autres poids calculés. Par opposées, j'entends que ces surfaces ne sont composées d'aucunes arêtes ayant pour l'une de leurs extrémités le point d'interpolation. Ceci est important pour l'extension de l'idée en 3D.

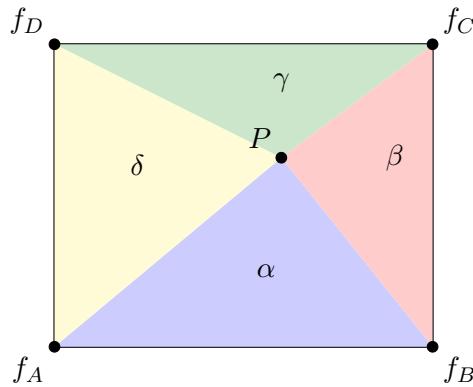


FIGURE 2.15 – Schéma d'interpolation dans un quadrilatère quelconque

Un point à souligner est qu'une multiplication de surfaces est réalisée, ce qui induit une équation d'ordre 4, ce qui pourrait impliquer une méthode du même ordre. Contrairement à son nom, l'interpolation bilinéaire est en réalité quadratique avec un résultat linéaire. Une vérification a été faite et la méthode n'interpolate pas sans erreurs les fonctions d'ordre supérieur à un. De plus, le quadratique ne peut pas englober pas le linéaire dans le cas où nous nous basons uniquement sur les quatre points d'un quadrilatère. Normalement il faudrait s'appuyer sur plus de points pour le quadratique. Finalement, la démonstration pourrait se construire en montrant la linéarité selon les axes x et y indépendamment (comme pour le bilinéaire). En effet, sur la figure 2.15 ci-dessous, nous remarquons que lors d'un déplacement selon l'horizontal, vers la droite, les surfaces α et γ ne varient pas tandis que la surface δ augmente linéairement et β diminue linéairement aussi.

Voici l'équation de la méthode d'interpolation linéaire pour le quadrilatère :

$$f(P) = \frac{f_A \cdot (\beta \times \gamma) + f_B \cdot (\gamma \times \delta) + f_C \cdot (\delta \times \alpha) + f_D \cdot (\alpha \times \beta)}{(\beta \times \gamma) + (\gamma \times \delta) + (\delta \times \alpha) + (\alpha \times \beta)} \quad (2.15)$$

3D

Pavé droit : Trilinéaire Pour le 3D, si le maillage est non-structuré, alors le type de cellule est le pavé droit. À ce moment, nous sommes dans le cas de l'interpolation dite trilinéaire. Encore une fois, la formule se trouve facilement dans la littérature et se retrouve facilement en étendant la formule de l'interpolation sur un rectangle. Nous associons comme poids à un des huit sommets s_1 le volume opposé, construit de la sorte :

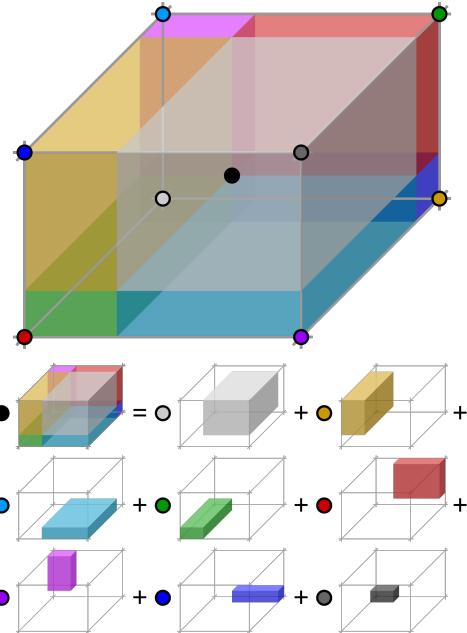


FIGURE 2.16 – Schéma d'interpolation trilinéaire (source : Wikipédia)

L'équation qui en découle est la suivante :

$$f(x, y, z) = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 f_{ijk} (1 - |x - x_i|)(1 - |y - y_j|)(1 - |z - z_k|) \quad (2.16)$$

Tétraèdre : Barycentrique étendu Pour le tétraèdre (une pyramide à base triangulaire quelconque), nous trouvons bien un volume opposé à chaque sommet. Opposé dans le sens où le sommet ne partage pas de segment avec le volume. Nous pouvons alors appliquer la même formule que pour le triangle, avec 4 volumes à la place de 3 surfaces.

Pyramide, hexaèdre, prisme et autres formes Pour la pyramide (à base quadrilatérale) et l'hexaèdre, une extension de la formule du quadrilatère sera faite. Il pourrait en être de même pour les prismes et autres formes, mais des problèmes d'implémentations apparaissent (voir ref 2.3.1).

Note pour l'ordre supérieur : une utilisation des maillages transfinis pourrait être faite [8].

2.3 Implémentation de la méthode linéaire

2.3.1 La structure générale du code "TreatmentInterpolation"

Le traitement d'interpolation d'Antares ne traite que des maillages ayant des valeurs uniquement au niveau des noeuds des cellules (pas dans les cellules ou entre les noeuds). Pour effectuer une recherche efficace des N points les plus proches de la base source par rapport à ceux de la base target, il utilise un k-dimensional tree (pour arbre à k-dimensions) (KDTree). Cette structure de données arborescente est conçue pour organiser des points dans un espace à k dimensions, ce qui est particulièrement utile pour effectuer des recherches rapides de voisins dans un ensemble de données multidimensionnelles. Le KDTree divise récursivement l'espace en sous-espaces plus petits en utilisant des plans perpendiculaires aux axes de coordonnées. À chaque niveau de l'arbre, les points sont séparés en fonction de l'une des dimensions, avec une alternance de dimensions à chaque niveau. Cette approche hiérarchique permet de partitionner l'espace de manière efficace, rendant la recherche des points voisins beaucoup plus rapide que si l'on comparait chaque point individuellement.

Après avoir recensé les différentes méthodes qui seraient applicables, ma seconde mission a été d'implémenter une interpolation linéaire dans Antares. Grâce à Nitrox, j'ai accès au code source de la librairie que je peux modifier. Le code 'interpolation.py' faisait environ 500 lignes. Il est orienté objet. Il prend comme arguments obligatoires la base source et la base cible et renvoie dans le cas le plus simple la base cible avec les valeurs interpolées. De manière simplifiée, dans le code, les zones de la base source sont fusionnées puis les instants sont parcourus. Cette fusion permet de faire un KDTree, qui permet concrètement de rechercher de manière efficace quels sont les N points de la base source les plus proches des points de la base cible (et les distances associées, utilisées dans la méthode 'idw'). Ensuite l'interpolation est calculée par l'une des deux méthodes 'principales' qui peut elle-même appeler d'autres fonctions ou méthodes d'Antares.

Nom	Val. défaut	Description
source	N/A	Base source
target	N/A	Base cible
coordinates	None	Le nom des variables de coordonnées spatiales
variables	None	Nom des variables à interpoler (toutes pas défaut)
tolerance	1e-10	Seuil de proximité pour utiliser la valeur du point le plus proche
duplicated_tolerance	None	Nbr de décimales d'arrondi pour la détection de points dupliqués (exact par défaut)
invdist_power	1	Paramètre de puissance p de la méthode IDW
nb_points	None	Paramètre de nombre de points N de la méthode IDW
with_boundaries	False	Utiliser ou non les données des limites

TABLEAU 2.1 – Arguments initiaux pour la méthode d'interpolation dans Antares

L'interpolation IDW Cette distance est définie à 1e-10 dans le traitement et peut être modifiée par l'utilisateur en ajustant l'argument 'tolerance' en ajoutant la ligne :

```
1 myt['tolerance'] = valeur
```

entre la ligne 2 et 5 de l'exemple présenté dans la partie '2.1 La librairie Antares'. Il en est de même pour les paramètres N et p avec pour argument respectivement 'nb_points' et 'invdist_power'.

Par défaut, dans le code, p est égal à '1' et N est égale au nombre de sommets de la première cellule de la liste de types de cellules de la base cible. Il y aurait peut-être une modification mineure à faire sur ce point. La structure de la base cible importe peu comparée à celle de la

base source, pour l'interpolation. Mais par souci de rétrocompatibilité, aucune modification ne doit changer l'utilisation qui a déjà été faite de ce traitement.

Une de mes missions était de chercher s'il y avait des paramètres plus optimisés que N et p pour cette méthode. Je n'ai pas trouvé la réponse dans les différents articles [11] et thèses que j'ai lues. C'est pour cela qu'il sera présenté plus tard dans le rapport comment des paramètres optimaux ont été trouvés en faisant des tests.

L'interpolation linéaire La méthode ne fonctionne pas pour tous les cas de connectivité prismatique, notamment pour les cellules constituées de triangles non parallèles entre elles dans l'espace et de surfaces différentes. De plus, l'ordre des sommets contenu dans la connectivité est différent en fonction du solveur dont est issue la base (par exemple dans hip elle est définie tel que sur l'image en annexe A3). Cela rend donc impossible (ou difficile) le calcul des volumes tel que la méthode a été implémentée. Nous avons effectivement besoin de savoir à quel sommet correspond quel indice. La connectivité prismatique ne sera donc finalement par traitée. Il en est de même pour les cellules plus complexes à N faces ou toute autre connectivité qui viendrait à apparaître. Cependant, il est simple de pallier ce problème. Il suffit de "tétraèdriser" la base source afin de créer des tétraèdres avant d'utiliser le traitement d'interpolation. Il en est de même pour de potentiels cas 2D non traités.

Voici comment réaliser cela :

```
1 import antares
2 from copy import deepcopy
3 # deepcopy est utilisé pour ne pas "tetrahedralize" la 'source_base'
4
5 copy_source_base = deepcopy(source_base)
6 # copy_source_base.unstructure() # Pour une base non structurée
7 t = antares.Treatment('tetrahedralize')
8 t['base'] = copy_source_base
9 copy_source_base = t.execute()
10
11 myt = antares.Treatment('interpolation')
12 myt['source'] = copy_source_base
13 myt['target'] = target_base
14 result2 = myt.execute()
```

Code 2.2 – Tetrahedralize pour interpoler linéairement tous types de cellules

Lorsque la base est structurée, il n'y a pas connectivité facilement accessible, donc pas d'interpolation linéaire implémentée. Il faut donc 'unstructure' la base source puis la "tétraèdriser". Cela s'effectue de la même façon que définie dans le code ci-dessus, en ajoutant

```
1 copy_source_base.unstructure()
```

à la ligne 6. Réellement, le structuré a été développé pour l'ordre un et deux, avec un bug pour l'ordre trois. Par manque de temps, la méthode n'a pas été finie d'implémenter. Étant rapide (généralement plus rapide que l'interpolation IDW), la fonctionnalité sera probablement disponible plus tard.

2.3.2 Le pseudo-algorithme

Voici le pseudo-algorithme simplifié donnant l'architecture de la fonction linéaire du traitement interpolation.

Initialisation :

- Créer le KD-tree
- Créer une liste de listes `node_to_elements` qui permet de récupérer l'indice des éléments constitués par un certain point.
- Définir les points à interpoler

Parcours des points à interpoler :

- Pour chaque point :
 - ◊ Si la distance au point le plus proche est inférieure à `tolerance` :
 - Le poids associé au `point cible` prend la valeur 1
 - Le sommet associé au `point cible` prend la valeur du `point source` le plus proche
 - ◊ Sinon :
 - Récupérer les indices des éléments des points les plus proches
 - Pour chaque élément pertinent :
 - Récupérer les coordonnées des sommets (points) de la cellule
 - ◊ Si le point est à l'intérieur de la cellule :
 - Calculer les poids 'barycentriques'
 - Récupérer les indices des sommets (points) de la base source
 - Si le point n'est dans aucune cellule proche :
 - Le poids associé prend la valeur `numpy.nan` (Cette valeur sera changée avant la mise à jour d'octobre)

Dans le cas où le maillage est le même entre tous les instants, ou que l'utilisateur a déjà récupéré les poids (et autres valeurs utiles) nous ne recalculons pas tous les paramètres, ce qui permet une diminution significative du temps de calcul.

Mise à jour de la base cible :

- Pour chaque variable à interpoler :
 - La variable de chaque nouveau point prend la valeur des poids calculés multipliés par la variable aux sommets (points) respectifs de la base source

2.3.3 Optimisation du temps de calcul

Plusieurs axes ont été exploités afin de diminuer le temps de calcul du traitement d'interpolation.

Le premier a été énoncé dans le pseudo-algorithme présenté dans la section précédente. Il s'agit de ne calculer qu'une seule fois le couple (poids, indices des points source) et les autres paramètres puis de les enregistrer. Cela est utile dans deux cas :

- Entre les instants si la structure des deux bases reste inchangée. C'est par exemple dans le cas des bases issues de solutions aéroacoustique. Dans mon cas test, la solution contient 200

instants, le temps de calcul est alors presque divisé par 200. Dans le code, un test est fait pour savoir si tous les instants sont partagés (voir la section 2.1). Si oui, alors les poids et autres données nécessaires sont calculés. Ensuite, tous les instants seront calculés avec les données. Cela permet d'éviter d'enregistrer les coordonnées une seule fois et ainsi réduire le coût de mémoire. Cette optimisation est faite automatiquement dans le traitement.

- Si le calcul sur la base source est réalisé plusieurs fois sans que sa structure soit changée, qu'elle ait un instant ou plusieurs instants partagés. Dans le cas où la structure du maillage est différente entre deux instants, si `get_data` a la valeur `True`, alors le programme renverra une erreur. Un cas où l'on pourrait imaginer utiliser cette fonctionnalité est lors de plusieurs simulations avec des résultats physiques différents entre les simulations, mais toujours sur le même maillage. Pour utiliser cette fonctionnalité, il faut légèrement changer l'appel à la fonction, voici un exemple :

```
1 import antares
2 from copy import deepcopy
3 # deepcopy est utilise pour s'assurer que la 'target_base' ne soit pas calculee
4           par le traitement
5
6 myt = antares.Treatment('interpolation')
7 myt['source'] = source_base
8 myt['target'] = deepcopy(target_base)
9 myt['get_data'] = True
10 result1, data = myt.execute()
11
12 # 'data' peut etre enregistre sur votre ordinateur par exemple, pour pouvoir
13   etre appele plus tard
14
15 myt = antares.Treatment('interpolation')
16 myt['source'] = source_base
17 myt['target'] = target_base
18 myt['data'] = data
19 result2 = myt.execute()
```

Code 2.3 – Exemple de réutilisation des données

Ici, `result2` aura été calculé beaucoup plus rapidement que `result1`. De plus, ce changement est aussi appliqué à la méthode IDW sans poser de problèmes de rétrocompatibilité, car les résultats sont inchangés. Pour le cas test d'aéroacoustique à 200 instants partagés, le traitement est cinq fois plus rapide pour l'IDW et plus de cent fois plus rapide pour la méthode linéaire.

La méthode linéaire était initialement 200 à 500 fois plus lente que IDW. En faisant du 'profiling' pour savoir quelles lignes de code étaient les plus coûteuses. Certaines listes ont été remplacées par des `np.arrays`, ce qui augmente la rapidité d'accès aux éléments de ce denier en mémoire. Le test de localisation d'un point dans une cellule et de calcul des poids a aussi été optimisé. Les surfaces créées par le point cible avec les sommets sont d'abord calculées, suivie par le calcul du volume de la cellule. Si ces volumes sont égaux à une erreur numérique près, alors le point est dans la cellule et les poids sont déterminés. Sinon, il faut nécessairement de chercher ailleurs.

Une autre idée a été testée pour réduire le temps de calcul. Elle reste linéaire, mais ne donne pas forcément le même résultat (lorsque la variable à interpoler est non-linaire, sinon le résultat reste la solution analytique). Cette idée est décrite uniquement pour le cas 2D par souci de simplicité de visualisation, mais est similaire pour le 3D.

Le principe est de ne pas calculer tout ce qui touche à la connectivité, car c'est ce qui augmente significativement le temps de calcul. Nous allons alors chercher les 3 points les plus proches (4 en 3D). En reliant les sommets, cela donnera une forme dite non croisée. Ensuite, nous faisons

le calcul des barycentres et appliquons la même méthode qu'au-dessus. La différence ici est que le point cible n'est pas forcément dans la cellule décrite par les 3 points les plus proches. Un point à noter est que si le point est sur une arête, il pourrait ne jamais être détecté. Alors un argument `tolerance_edge` a été créé. Si la tolérance relative entre la somme des sous-surfances créées par le point p et la surface totale de la cellule est inférieure à `tolerance_edge`, alors le point est considéré comme dans la cellule.

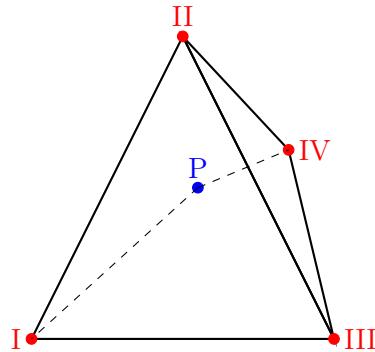


FIGURE 2.17 – Schéma de sélection du deuxième point le plus proche

Alors nous pourrions avoir une ou des surfaces formées par le point cible qui sortent du triangle initial (et la somme des surfaces ne serait pas égale à la surface du triangle initial). Mais pour respecter la linéarité, il suffit de faire le même calcul en prenant en compte le fait que la ou les deux surface(s) entièrement extérieures à la cellule sont négatives. La figure ci-dessous (Figure 2.18) est un zoom sur le schéma 2.17.

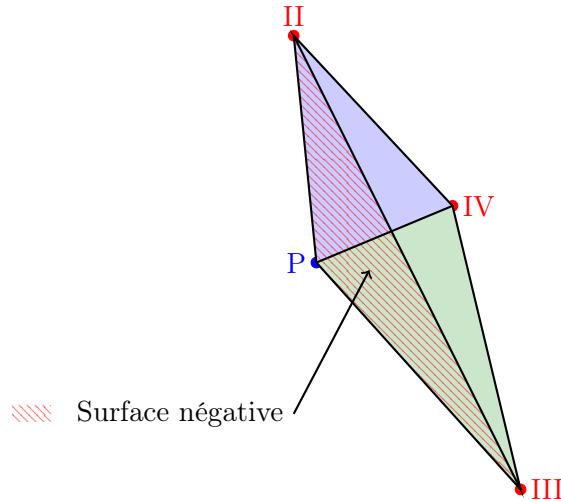


FIGURE 2.18 – Illustration de la surface négative avec le point P

N'ayant pas trouvé une façon simple de savoir si la surface était négative, par manque de temps et crainte de l'ajout de complexité (ce qui augmente le temps de calcul), cette méthode n'a pas été implémentée. Cela permet aussi une extrapolation linéaire sans créer d'erreur.

Finalement, voici les arguments ajoutés pour la méthode linéaire :

Nom	Val. défaut	Description
tolerance_edge	1e-10	Tolerance 'point dans la cellule ?'
method	'idw'	Méthode d'interpolation ('idw' ou 'linear')
data	None	Mettre les données précalculées
get_data	None	True pour récupérer des données

TABLEAU 2.2 – Arguments ajoutés pour la méthode d'interpolation dans Antares

2.4 Tests

A présent, le but est de savoir quelle est la meilleure méthode, notamment dans le cas de l'interpolation pour l'application au traitement FWH. Pour cela il faut tester si la méthode linéaire fonctionne correctement, et tester les meilleurs paramètres N et p de la méthode IDW.

L'objectif d'améliorer la précision de l'interpolation est d'avoir des résultats plus précis et/ou de diminuer le coût de calcul en déraffinant le maillage source. Comme l'explique l'article [29], le raffinement du maillage source est clef pour l'interpolation (et ensuite les résultats aéroacoustique qui en découlent).

2.4.1 Tests unitaires sur la méthode linéaire

Pour l'instant, je n'ai démontré que par l'expérimentation que la méthode implémentée était linéaire.

Le premier test, créé en même temps que l'implémentation dans l'objectif de tester si la méthode fonctionne dans un cas simple, est de créer une variable linéaire et de faire une interpolation sur un second maillage. L'idée est de créer une base source avec une variable linéaire et une base cible avec des points décalés. Ensuite, on teste l'interpolation renvoie bien la valeur analytique espérée sur tous les types de cellules. Ce test est effectué simplement avec une commande 'pytest' dans le terminal. En comparant les valeurs analytiques et numériques, l'erreur observée est nulle. Ce test est par ailleurs implémenté (pour la connectivité linéaire, triangulaire, quadrilatérale, tétraédrique et hexaédrique) dans les tests unitaires d'Antares, qui permettent de savoir que la librairie fonctionne correctement, notamment lors d'un git push vers Nitrox. Cependant, il est plus difficile de créer de manière simple un maillage avec des pyramides ou tétraèdres en python. Une vue en coupe sur une base avec ces deux types de cellules a été utiliser pour compléter le test du code d'interpolation linéaire. La base contiens une variable linéaire, et nous vérifions bien que l'erreur est nulle. Voici la vue une visualisation (sur le logiciel Paraview) de l'interpolation linéaire Figure 2.19 (parfaitement lisse) que nous avons aussi comparé avec l'interpolation IDW Figure 2.20 qui s'avère moins "lisse".

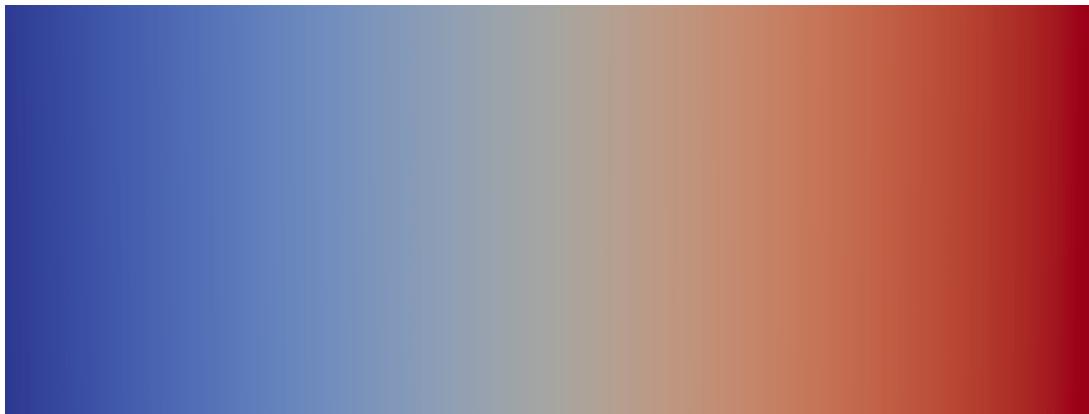


FIGURE 2.19 – Vue en coupe sur des cellules pyramidales et tétraédriques par interpolation linéaire

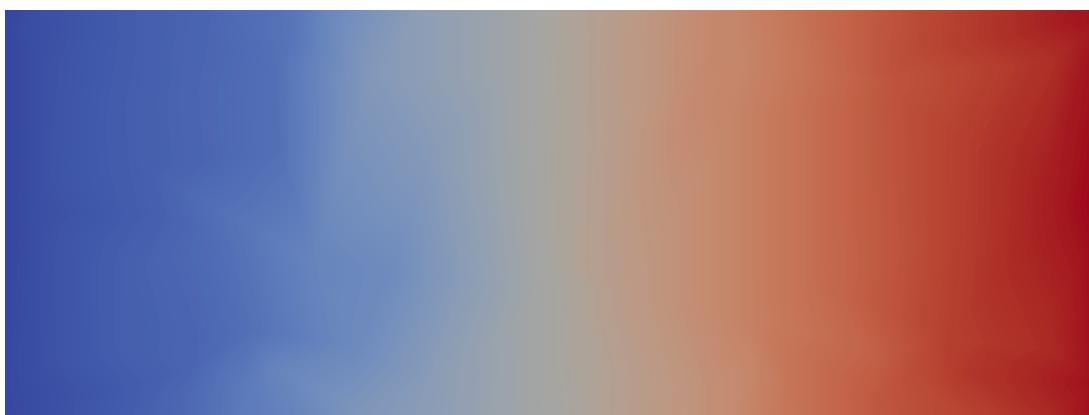


FIGURE 2.20 – Vue en coupe sur des cellules pyramidales et tétraédriques par interpolation idw

Ensuite, j'ai essayé de tracer le log de l'erreur au sens des moindres carrés en fonction du raffinement du maillage source. Cela a étonnamment montré une convergence d'ordre quatre pour tous types de cellules (alors que nous nous attendions à une convergence d'ordre un pour le 1D). En voici l'exemple pour le 1D (Figure 2.21) (pour les autres types de cellules, les courbes sont similaires).

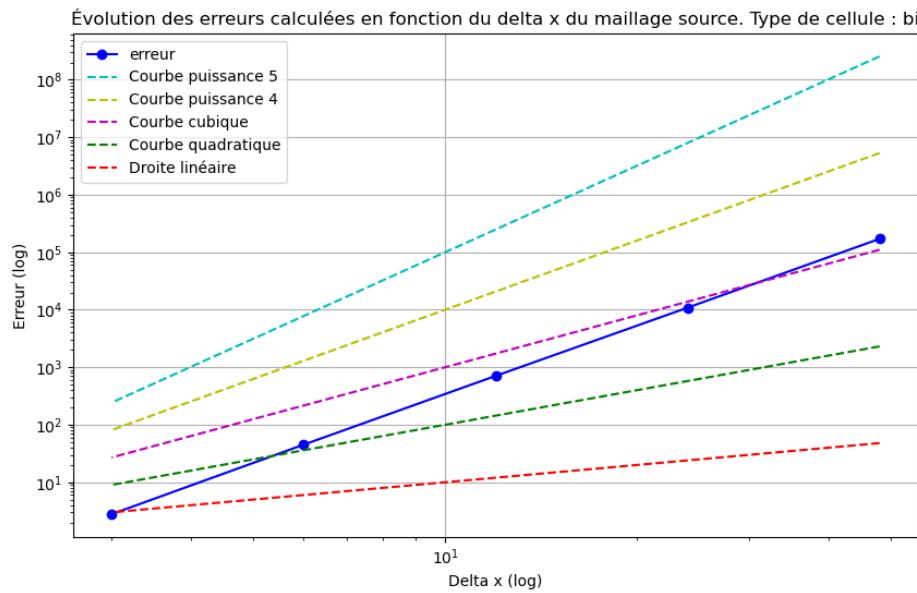


FIGURE 2.21 – Log de l'erreur en fonction du log du raffinement

2.4.2 Tests sur des cas industriels

Pour tester la robustesse du traitement (notamment dans les cas multizones, multiconnexités et multi-instants), son application à des tests industriels a été testée. Voici deux images (Figure 2.22 et Figure 2.23) comparant l'interpolation IDW et linéaire sur un maillage 3D solution d'une flamme dans une chambre de combustion en base source et un plan en base cible qui coupe la base source en $(0, 0, 0.01)$, selon l'axe z.

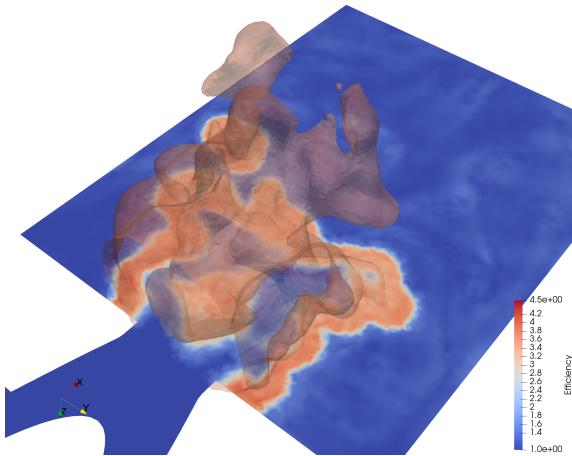


FIGURE 2.22 – Interpolation IDW sur une coupe d'une chambre de combustion avec iso-surface de la flamme

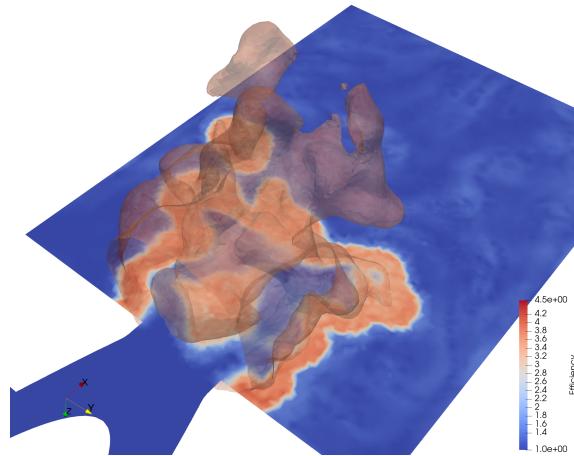


FIGURE 2.23 – Interpolation linéaire sur une coupe d'une chambre de combustion avec iso-surface de la flamme

Nous observons bien que les résultats sont similaires dans les deux cas (avec de légères différences démontrant que les méthodes sont bien différentes).

Voici les résultats de temps de calcul et d'erreur avec l'interpolation du logiciel vtk :

- Interpolation linéaire : 16.634 s, MSE : 1.7e-28
- Interpolation IDW : 0.661 s, MSE : 4.0e-03

- Rapport de temps de calcul linéaire/IDW : 25

L'erreur au sens des moindres carrés est quasiment nulle pour l'interpolation linéaire, nous en déduisons que vtk interpole aussi linéairement. L'erreur numérique correspond probablement à une méthode d'interpolation linéaire différente utilisée par vtk mais le résultat théorique est le même.

2.4.3 Tests sur des cas d'aéroacoustique

Le traitement FWH permet de déterminer le son à grande distance émanant d'une source aéroacoustique. Cela peut être utilisé pour déterminer numériquement le bruit d'un moteur d'avion par exemple. Dans notre cas test, la source n'est pas un moteur, mais une onde aéroacoustique simulé dans un cube. Voilà comment fonctionne notre chaîne aéroacoustique. Nous créons une base source rectangulaire, sur cette image (Figure 2.24a) elle est constituée de 3×5 nœuds et 4^3 éléments. Nous lui appliquons une variable sinusoïdale représentant la variation de pression en fonction du temps. Nous créons en parallèles une base cible sphérique (Figure 2.24b), représentant la surface que nous créons dans le maillage de la solution englobant l'objet étudié (le moteur par exemple). Ensuite, nous interpolons la solution de la base source sur la surface cible (Figure 2.24c). Les équations FWH sont appliquées sur cette surface. Elles permettent ensuite d'avoir la solution analytique de l'onde aéroacoustique à une distance éloigné de la cible. Nous appliquons l'équation pour un point de l'espace pour obtenir les variations de pression, impliquant du son (à différentes fréquences et amplitudes).

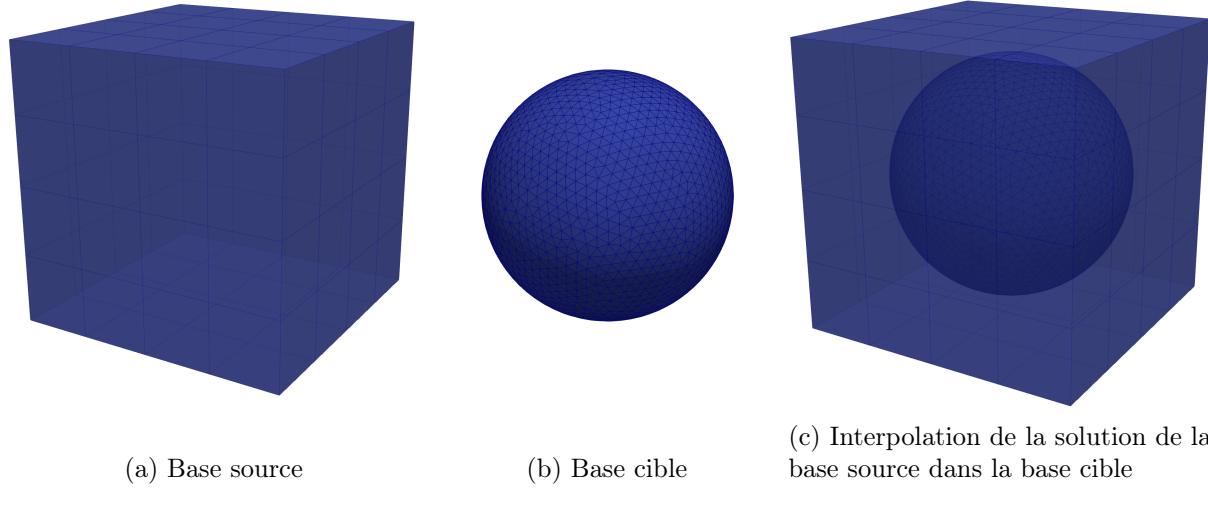
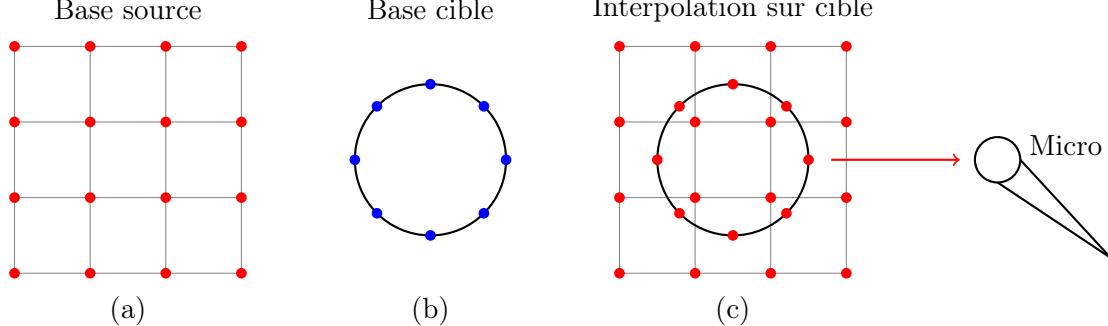


FIGURE 2.24 – Début de la chaîne aéroacoustique



La base source contient la pression, aéroacoustique, d'une onde sinusoïdale dans un maillage grossier (différents types de cellules ont été testés). La base cible est une sphère d'où les équations de FWH seront ensuite appliquées. Finalement, voilà le résultat (Figure 2.25) pour ces paramètres :

- observateur quelconque, car Mach égal à 0 ;
- fréquence du signal aéroacoustique : 40 Hz ;
- nombre de points de la base source : 3x4 (pour 3D et 4 points par dimension) ;
- types de cellules de la base source : hexaèdres ;
- paramètres N et p par défaut ;
- longueur de la base source : 7,20 m ;
- diamètre de la base cible : 6 m ;
- gamma = 1.4 ;
- vitesse du son : 340.0 m/s ;
- masse volumique de l'air = 1.1825 kg/m³ ;
- constante spécifique de l'air = 287.058 m²/(s²xK).

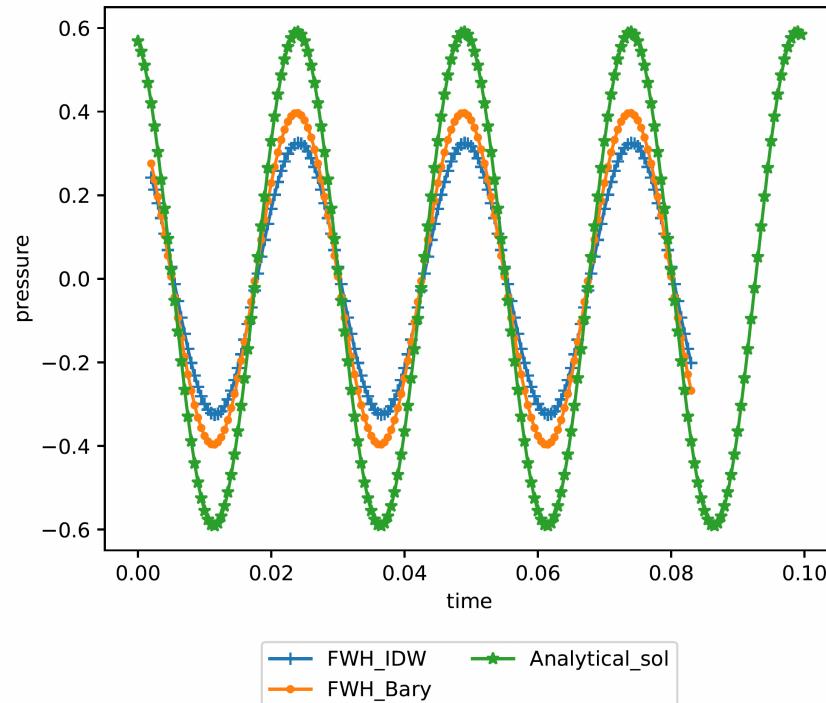


FIGURE 2.25 – Comparaison entre la solution analytique et la solution utilisant l'interpolation IDW et linéaire

L'erreur, au sens des moindres carrés, est plus élevée pour la méthode IDW que pour la méthode linéaire (ce qui est observable sur la Figure 2.25).

Une amélioration possible de l'interpolation pour le cas de l'aéroacoustique serait d'utiliser la méthode d'interpolation par partie décrite dans des articles de l'ONERA [9] [10]

Pour des cas test et applications dédiés à l'aéronautique, il peut être intéressant de regarder les caractéristiques du bruit d'un avion, comme ses fréquences démission dans l'audible. [28]

2.4.4 Tests sur les paramètres de la méthode IDW

La méthode IDW contient deux paramètres (N et p) dans son équation présentée plus haut.(ref 2.3.1)

Pour savoir quels paramètres sont optimaux, des tests ont été menés sur le même cas que dans la sous-partie précédente, mais cette fois, ce qui nous intéresse est l'erreur au sens des moindres carrés, entre la solution analytique et IDW, en fonction des paramètres (N et p). Cette erreur est liée au rapport d'amplitude entre les deux courbes, ce qui a finalement été pris en compte. En les faisant varier, nous trouvons un minimum (local) pour $N=10$ et $p=10$ pour des paramètres similaires à la figure 2.25. Ces tests ont été menés sur kraken afin de ne pas perdre trop de temps.

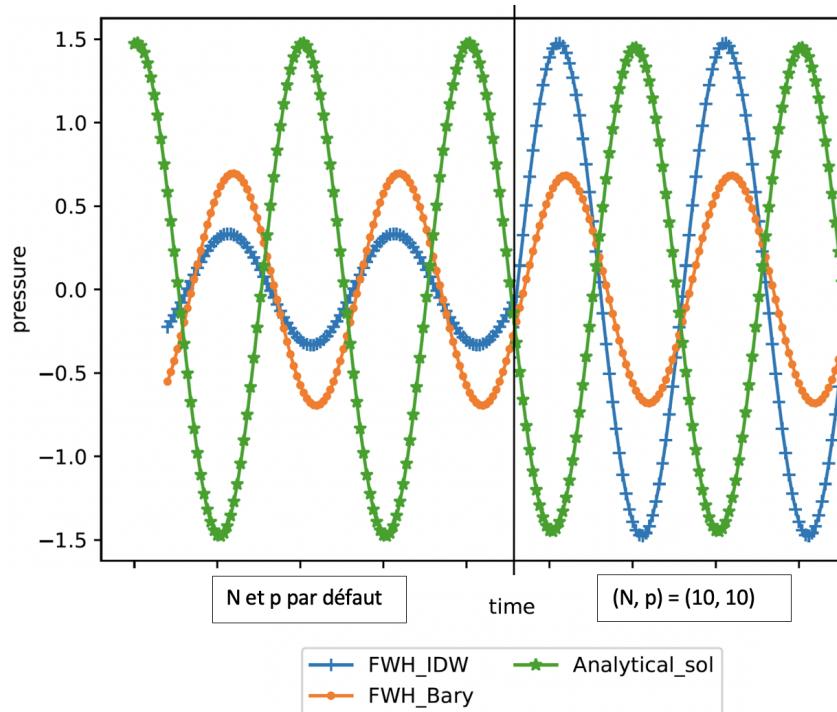


FIGURE 2.26 – Comparaison entre la solution analytique et la solution utilisant l'interpolation IDW ($p=10$, $N=10$) et linéaire

À gauche, la méthode linéaire est utilisée, à droite la méthode IDW avec comme paramètres ($N=10$, $p=10$)(2.26). Le décalage latéral des courbes est un artefact (et cela permet de mieux distinguer les courbes bleues et vertes dans notre cas.) Le rapport d'amplitude avec la solution analytique est le suivant : idw = 102%, linéaire = 48%. Soit une erreur de 2% pour idw et 52% pour le linéaire.

Conclusion

Bilan technique et humain

Ce stage au CERFACS m'a vraiment plu. Il m'a permis de développer et renforcer beaucoup de connaissances dans le milieu de l'informatique, mais aussi sur le plan humain. J'ai pu découvrir comment se déroulait la vie en laboratoire de recherche.

L'objectif de ce stage était d'implémenter une méthode d'interpolation linéaire dans Antares puis de comparer sa précision avec la méthode IDW dans le cas de propagation acoustique avec l'analogie FWH, tout en optimisant la vitesse d'exécution du code. Après avoir exploré la structure d'Antares, une recherche documentaire a été menée pour savoir quelle méthode, implémentable dans la librairie, pouvait être candidate pour compléter la méthode idw et linéaire. Quelques méthodes polynomiales et géostatistiques en sont ressorties. Une présentation un plus détaillé de la méthode par voisin le plus proche, IDW et linéaire est aussi donné. L'un des principaux résultats de ce stage a été l'implémentation globalement réussie de la méthode d'interpolation linéaire, qui s'est avérée généralement plus précise que la méthode de Pondération Inverse à la Distance (IDW) précédemment codée. Coder l'interpolation linéaire jusqu'en 3D n'a pas été très difficile ou chronophage, mais bien l'intégrer au code déjà existant et faire les tests l'était plus. Heureusement que j'avais le soutien de mon maître de stage pour tout cela. Cette nouvelle méthode a été testée sur des cas académiques comme sur des cas industriels comme sur une vue en coupe à la position (0, 0, 0.01), selon l'axe z de la chambre de combustion ///nom///. Ce travail aussi amélioré les performances du code (cinq fois plus rapide pour la méthode IDW dans un cas test d'aéroacoustique). Le code ajouté pourrait aussi aider à l'intégration de méthodes d'ordre supérieur grâce à certaines fonctions python réutilisables. J'ai ressenti tout au long de mon stage l'importance de faire un travail qui puisse être continué plus tard par un autre chercheur. Enfin, ce stage a été une expérience enrichissante qui a conforté mon intérêt pour les méthodes numériques appliquées et la recherche en calcul scientifique. Je suis reconnaissant pour l'accompagnement que j'ai reçu tout au long de cette expérience et pour les opportunités de développement personnel et professionnel qu'elle m'a offert.

Perspectives

Pour continuer à améliorer le traitement d'interpolation d'Antares, on pourrait implémenter l'une des méthodes d'ordre supérieur citées dans la section 2.2.3. Afin d'améliorer la rapidité, le code pourrait être passé en parallèle pour pouvoir s'exécuter sur plusieurs processeurs à la fois. Une petite amélioration qui permettrait de rendre le code compatible avec plus de base serait d'inclure les bases 'multi-zones' à l'interpolation linéaire.

Réflexion personnelle

Le CERFACS est un laboratoire privé à la pointe des simulations numériques. Il s'est d'abord spécialisé dans l'aérodynamique et s'intéresse maintenant de plus en plus à la combustion. Il adapte ses domaines de recherche vers les domaines importants de l'industrie. Durant ce stage,

au-delà des aspects techniques, j'ai eu un bel aperçu du métier de chercheur tant au travers de mes missions que dans mes rencontres. Il s'agit de rechercher et de s'appuyer sur le travail déjà réalisé autour de notre problématique. Ensuite, il faut essayer des pistes qui peuvent s'avérer infructueuses après les avoirs longuement explorés. Finalement un travail de rédaction est nécessaire pour expliquer clairement notre travail à la science.

Annexes

Les principaux postes d'émissions identifiés	Quelques objectifs et actions
Missions	- Prévention sur le sujet sur le site Carbon Footprint du Cefacs et affiches présentes dans les locaux.
Chauffage	- Rénovation du circuit d'alimentation en eau glacée des ventilo-convection de l'ancien bâtiment (Actions 2023 et 2024) - Traitement de l'étanchéité et de l'isolation de l'ancien bâtiment (analyse réalisée, actions en cours réparties sur plusieurs années pour cause de coût global).
Calculateurs internes et usage	- Un calculateur a été arrêté en février 2023 et remplacé seulement en 2024. - Une sensibilisation à l'optimisation de l'usage des calculateurs
Calculateurs internes et fabrication	
Électricité hors clusters	- Automatisation de l'éclairage des circulations de l'ancien bâtiment (abandon de l'éclairage manuel).
Trajet domicile-travail	- Participation à des initiatives en faveur du vélo ("Deux Pieds Deux Roues - 2P2R", "Objectif Employeur Pro Vélo") - Mise en place du Forfait Mobilité Durable pour le vélo. - Mise à disposition de deux vélos pour le personnel du Cefacs
Fuites de fluides	
Calculateurs externes et usage	- Une sensibilisation à l'optimisation de l'usage des calculateurs
Calculateurs externes et fabrication	
Matériel informatique	Les postes de travail sont globalement à jour et ont un bon niveau de technologie (77% publiés en 2023).

TABLEAU A3 – Axes de travail et exemples d'actions pour limiter l'empreinte carbone du CERFACS

Axes de travail	Exemples d'actions
Optimiser l'organisation et gestion du travail au niveau global	<ul style="list-style-type: none"> - Organisation d'une réunion générale du Cerfacs par la direction 2 fois par an - Valorisation de la bibliothèque (réaménagement de l'espace) - Amélioration de la communication interne (journal de la QVT disponible sur l'intranet, newsletter interne mensuelle) et externe (nomination d'un référent Communication)
Optimiser l'organisation et gestion du travail au niveau de l'équipe	<ul style="list-style-type: none"> - Proposer une formation à l'organisation et à la tenue de réunions efficaces
Optimiser l'organisation et gestion du travail au niveau personnel	<ul style="list-style-type: none"> - Information/sensibilisation au burn-out (par la médecine du travail)
Assurer un meilleur accueil et support des non permanents	<ul style="list-style-type: none"> - Mise en place d'un groupe de travail pour optimiser l'encadrement des non-permanents (doctorants, post-doctorants...) - Rédaction d'une charte QVT
Favoriser la vie sociale	<ul style="list-style-type: none"> - Organisation d'événements sociaux en dehors du temps de travail - Aménagement d'une salle de repos - Organisation de pauses café collectives mensuelles
Améliorer le confort matériel	<ul style="list-style-type: none"> - Nouvelle machine à café à grain mise à disposition pour tous (avec l'achat de café par le Cerfacs) - Gourdes métalliques offertes à l'ensemble du personnel - Achat et installation de nouveaux arceaux pour augmenter la capacité d'accueil des vélos - Sensibilisation à l'ergonomie sur le poste de travail (par la médecine du travail)

TABLEAU A4 – Axes de travail et exemples d'actions pour la Qualité de Vie au Travail au CERFACS

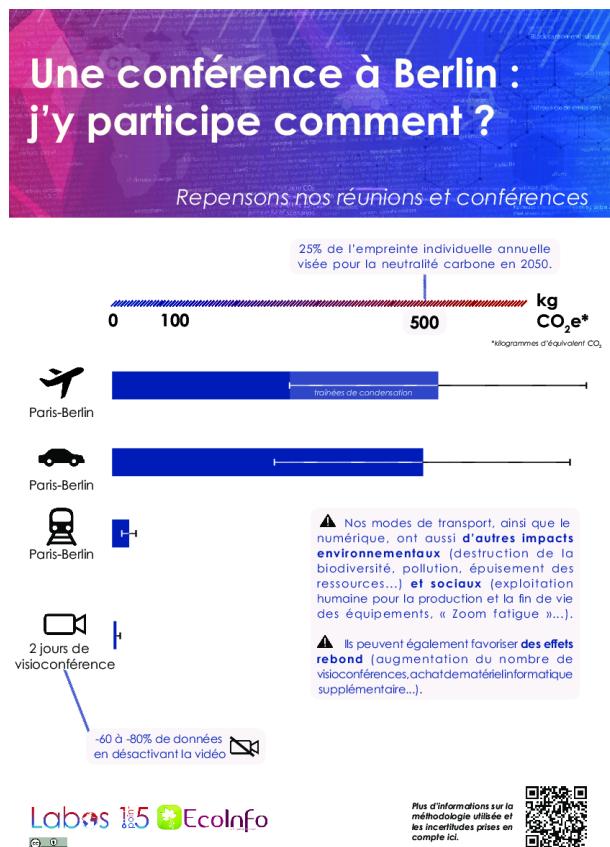


FIGURE A1 – Empreinte carbone s'une conférence à Berlin

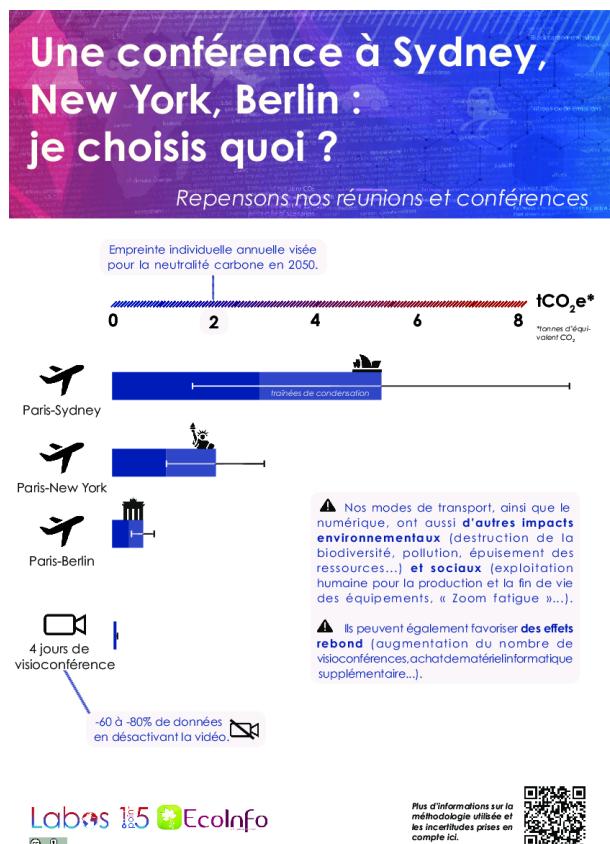


FIGURE A2 – Empreinte carbone s'une conférence à Sydney

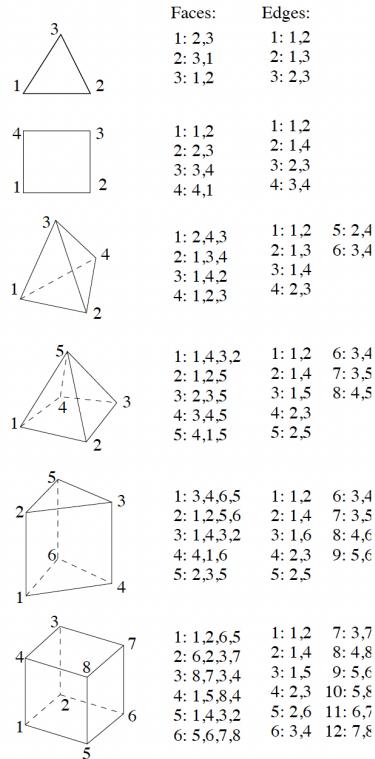


Figure 2: hip's canonical elements.

FIGURE A3 – Connectivité dans hip

Bibliographie

- [1] ALEXIS BOUDIN. « Numerical Methods for Large Eddy Simulation in Turbomachinery (en cours) ». Thèse de doct. Soutenance prévue en 2025.
- [2] ANTARES DEVELOPMENT TEAM. *Antares Documentation Release 2.2.0*. Avr. 2024. URL : <https://cerfacs.fr/antares/>.
- [3] Chandrajit BAJAJ. « Multi-dimensional Hermite Interpolation and Approximation for Modelling and Visualization ». In : *ICCG* (2001).
- [4] C BENOIT et al. « Cassiopee : a CFD pre-and post-processing tool ». In : *Aerospace Science and Technology* (2015). DOI : [10.1016/j.ast.2015.05.023](https://doi.org/10.1016/j.ast.2015.05.023).
- [5] Jean-Paul BERRUT et Lloyd N. TREFETHEN. « Barycentric Lagrange Interpolation ». In : *SIAM Review* (2004). DOI : [10.1137/S0036144502417715](https://doi.org/10.1137/S0036144502417715).
- [6] BITBUCKET. *RBF and Kernel Interpolations*. Accédé : 2024-08-22. 2021. URL : <https://bitbucket.org/joelrosenfeld/rbf-and-kernel-interpolation/src/master/>.
- [7] CADENCE CFD. *The Best Methods for Mesh Interpolation*. Accédé : 2024-08-23. URL : <https://resources.system-analysis.cadence.com/blog/msa2022-the-best-methods-for-mesh-interpolation>.
- [8] Ricardo CAMARERO. *MEC6212 : Génération de maillages transfinis*. 2024.
- [9] Guilherme CUNHA et Stephane REDONNET. « An Innovative Interpolation Technique for Aeroacoustic Hybrid Methods ». In : *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA Aeroacoustics Conference)*. 2011. DOI : [10.2514/6.2011-2754](https://doi.org/10.2514/6.2011-2754).
- [10] Guilherme CUNHA et Stephane REDONNET. « Development of optimized interpolation schemes with spurious modes minimization ». In : *International Journal for Numerical Methods in Fluids* (2016). DOI : [10.1002/fld.4079](https://doi.org/10.1002/fld.4079).
- [11] ESRI. *How IDW (Inverse Distance Weighted) Interpolation Works*. Accédé : 2024-08-13. 2024. URL : <https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-idw-works.htm>.
- [12] FERRITE. *interpolation.jl*. Accédé : 2024-08-22. 2024. URL : <https://github.com/Ferrite-FEM/Ferrite.jl/blob/master/src/interpolations.jl>.
- [13] FL FLUIDSSENGINEER, FL MARCELLILIE et FL PAULLSCHALLHORN. « Interpolation Method Needed for Numerical Uncertainty Analysis of Computational Fluid Dynamics ». In : *52nd Aerospace Sciences Meeting*. 2014. DOI : [10.2514/6.2014-1433](https://doi.org/10.2514/6.2014-1433).
- [14] GETFEM. *Finite element method list*. Accédé : 2024-08-22. URL : <https://getfem.readthedocs.io/en/latest/userdoc/appendixA.html>.
- [15] William J GORDONT. « Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation ». In : *SIAM Journal on Numerical Analysis* (1971). DOI : [10.1137/0708019](https://doi.org/10.1137/0708019).
- [16] William J GORDONT. « Blending-function methods of bivariate and multivariate interpolation and approximation ». In : *International journal for numerical methods in engineering*, Vol. 7 (1971).

- [17] Curtis GROVES, Marcel ILIE et Paul SCHALLHORN. « Interpolation Methodneeded for Numerical Uncertainty Analysisof Computational FluidDynamics ». In : NASA (2012). URL : <https://ntrs.nasa.gov/api/citations/20140011325/downloads/20140011325.pdf>.
- [18] Kai HORMANN. « Barycentric Interpolation ». In : (). URL : <https://www.inf.usi.ch/hormann/papers/Hormann.2014.BI.pdf>.
- [19] JOEL ROSENFIELD. *The Learning Dock*. Accédé : 2024-08-26. 2024. URL : <https://www.thelearningdock.org/>.
- [20] J. de LABORDERIE et al. « Numerical analysis of a high-order unstructured overset grid method for compressible LES of turbomachinery ». In : *Journal of Computational Physics* (2018). DOI : [10.1016/j.jcp.2018.02.045](https://doi.org/10.1016/j.jcp.2018.02.045).
- [21] David LEVIN. « Mesh-Independent Surface Interpolation ». In : *Geometric modeling for scientific visualization*. 2004. DOI : [10.1007/978-3-662-07443-5_3](https://doi.org/10.1007/978-3-662-07443-5_3).
- [22] MAPSCAPING. *Understanding Inverse Distance Weighting*. Accédé : 2024-08-13. 2023. URL : <https://mapscaping.com/understanding-inverse-distance-weighting/>.
- [23] « Mesh-Independent Surface Interpolation ». In : *Geometric Modeling for Scientific Visualization*. 2004. DOI : [10.1007/978-3-662-07443-5_3](https://doi.org/10.1007/978-3-662-07443-5_3).
- [24] Jens-Dominik MÜLLER. *A user's guide to hip*. 2020.
- [25] Grant E PALMER. « Construction of CFD Solutions Using Interpolation Rather than Computation with the ADSI Code ». In : *47th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition* (2009). DOI : [10.2514/6.2009-141](https://doi.org/10.2514/6.2009-141).
- [26] Alain PERRONNET. « Interpolation transfinie SW le triangle, le tétraèdre et le pentahèdre. Application à la création de maillages et à la condition de Dirichlet ». In : *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* (1998). DOI : [10.1016/S0764-4442\(97\)82723-1](https://doi.org/10.1016/S0764-4442(97)82723-1).
- [27] Les PIEGL et Wayne TILLER. *The NURBS Book*. 1995. DOI : [10.1007/978-3-642-97385-7](https://doi.org/10.1007/978-3-642-97385-7).
- [28] Christophe ROSIN, Bertrand BARBO et Boris DEFREVILLE. « Monitoring du bruit des avions : une détection à partir du signal audio ». In : (2010). URL : <https://hal.science/hal-00538359/file/000137.pdf>.
- [29] Stefan J. SCHODER et al. « Conservative source term interpolation for hybrid aeroacoustic computations ». In : *25th AIAA/CEAS Aeroacoustics Conference, 2019*. 2019. DOI : [10.2514/6.2019-2538](https://doi.org/10.2514/6.2019-2538).
- [30] SCIENCEDIRECT. *Kriging*. Accédé : 2024-08-22. URL : <https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/kriging>.
- [31] Nobuatsu TANAKA, Toshiteru YAMASAKI et Takaya TAGUCHI. *Accurate and Robust Fluid Analysis Using Cubic Interpolation with Volume/Area Coordinates (CIVA) Method on Unstructured Grids* *. 2004. DOI : [10.1299/jsmeb.47.672](https://doi.org/10.1299/jsmeb.47.672).
- [32] D. THIERRY. « interpolation dans un maillage et visualisation par isovaleurs ». In : *mode d'emploi du logiciel INGRID* (1986).

Liste des acronymes

1D une Dimension

AAM Advanced Aerodynamics and Multiphysics (pour Aérodynamique avancée et multi-physique)

ALGO-COOP parallels ALGORithmes & sCientifics sOftware Operational Performances (pour Algorithmes parallèles et logiciels scientifiques performances opérationnelles)

CAO Conception Assistée par Ordinateur

CERFACS Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique

CNES Centre National d'Études Spatiales

CSE Comité Social et Économique

CSG Computer Service General (pour Équipe informatique et support utilisateur)

EDF Électricité De France

ES Energy and Safety (pour Énergie et sécurité)

FWH Ffowcs Williams-Hawkins

GIP Groupement d'Intérêt Public

GLOBE modelling climate and GLOBal Change (pour Modélisation du climat et de son changement global)

HPC High Performance Computing (pour calcul haute performance)

IDW Inverse Distance Weighting

KDTree k-dimensional tree (pour arbre à k-dimensions)

MISCOG Multi-Instance Solver Coupled trough Overlapping Grids (pour Solveur multi-instances couplé à travers des grilles superposées)

MLS Moving Least Squares (pour Moindres carrés mobiles)

NURBS Non Uniform Rational Basis Splines (pour B-splines rationnelles non uniformes)

ONERA Office National d'Études et de Recherche Aérospatiales

OTP One-Time-Password

QVT Qualité de Vie au Travail

RBF Radial Basis Function (pour fonction de base radiale)

RGPD Règlement Général sur la Protection des Données

RSE Responsabilité Sociétale des Entreprises

SAS Société par Actions Simplifiées

Résumé L'objectif de ce stage était d'implémenter une méthode d'**interpolation linéaire** dans **Antares** puis de comparer sa précision avec la **méthode IDW** dans le cas de propagation acoustique avec l'analogie FWH, tout en optimisant la vitesse d'exécution du code. Premièrement, une recherche documentaire sur les méthodes d'interpolation implémentables dans notre cas a été menée. Ensuite, la méthode linéaire a été implémentée en utilisant une méthode particulière pour les cellules non triangulaires ou rectangulaires. L'efficacité a été augmentée dans les cas multi-instants partagés en évitant le recalcule des coefficients d'interpolation à chaque itération. Ces mêmes coefficients peuvent être récupérés par l'utilisateur qui voudrait appeler deux fois le traitement pour une base ayant une même structure. Le code est jusqu'à n fois plus rapide pour une base ayant n instants partagés. La méthode linéaire a été testée sur des cas simples et industriels. Elle fonctionne correctement sur tous les types de cellules, sauf dans certains cas très complexes comme des prismes ayant des faces opposées de tailles différentes et/ou non parallèles. Dans le cas de l'aéroacoustique, la méthode linéaire est plus précise que IDW dans tous les cas testés. Cependant, des résultats expérimentaux ont montré que des coefficients (N, p) autour de (10, 10) donnaient de très bons résultats. Dans ce cas, IDW peut s'avérer plus précis que la méthode linéaire.

Abstract The objective of this internship was to implement a **linear interpolation** method in **Antares** and then to compare its accuracy with the **IDW method** in the case of acoustic propagation using the FWH analogy, while optimizing the code execution speed. First, a literature review on interpolation methods applicable to our case was conducted. Secondly, the linear method was implemented using a special method for non-triangular or rectangular cells. Efficiency was increased in shared multi-instant cases by avoiding the recalculation of interpolation coefficients at each iteration. These same coefficients can be reused by a user who wants to run the process twice on a base with the same structure. The code is almost n times faster for a database with n shared instants. The linear method has been tested on simple and industrial cases. It works correctly on all cell types, except in some very complex cases, such as prisms with non-parallel and/or differently sized opposing faces. In the case of aeroacoustics, the linear method is more accurate than IDW in all tested scenarios. However, experimental results have shown that coefficients (N, p) around (10, 10) produce very good results. In such cases, IDW may prove to be more accurate than the linear method.