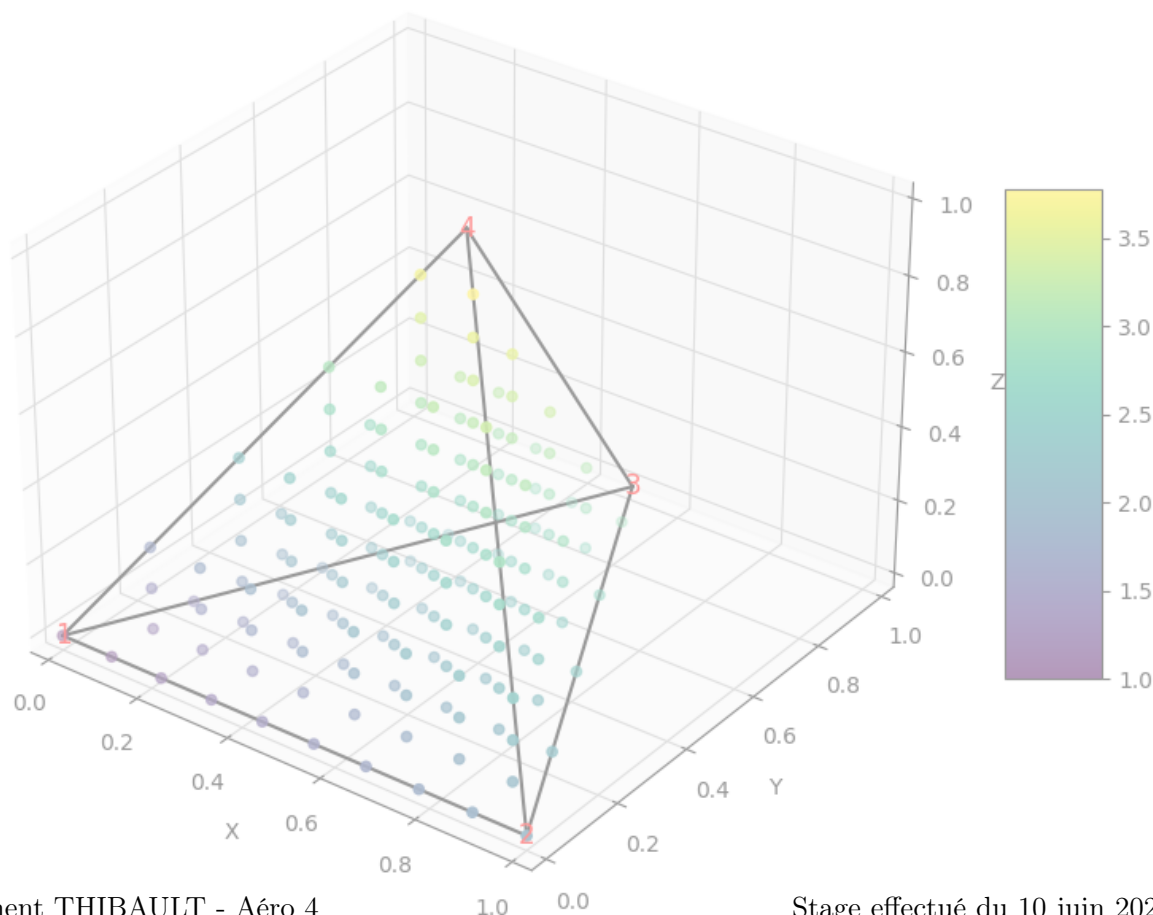


Rapport de stage Aéro 4

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 en 3D dans un tétraèdre



Remerciements

Je tiens à sincèrement remercier mon maître de stage Carlos pour m'avoir guidé pendant 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 de faire ce stage.

Merci pour l'aide précieuse en mathématiques de mes amis montagnards et matheux Benjamin CANOVAS-ANDRIEUX et Dimitri LANIER ainsi que celle de mon professeur de mathématiques à l'IPSA, Guillaume COUFFIGAL.

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, à mon ami, colocataire, collègue au CERFACS et à l'IPSA, Kélian RENOUX, à Arthur COLOMBIÉ, à Guillaume DAVILLER, à l'administration, au CSG et à Luc POTIER, mon cobureau, pour l'aide qu'ils m'ont apportée au CERFACS.

Table des matières

Bilan Technique	4
Introduction	5
1 Le CERFACS	6
2 Présentation du stage	8
2.1 La librairie Antares	8
2.2 Les différentes méthodes d'interpolation	9
2.2.1 L'interpolation par voisin le plus proche	9
2.2.2 L'interpolation Inverse Distance Weighting (IDW)	10
2.2.3 L'interpolation polynomiale	13
2.2.4 L'interpolation par Splines	13
2.2.5 Méthodes géostatiques	13
2.2.6 Méthode par moindres carrés	13
2.2.7 MISCOG	13
2.2.8 L'interpolation linéaire	13
2.2.9 Résumé des similitudes et différences des différentes méthodes	15
2.3 Implémentation de la méthode trilinéaire	15
2.3.1 La structure générale du code TreatmentInterpolation	15
2.3.2 Le Pseudo-algorithme	16
2.3.3 Optimisation du temps de calcul	16
2.3.4 Le résultat	17
2.4 Tests sur des cas d'aéroacoustique	17
2.4.1 Tests sur les paramètres de la méthode IDW	17
2.4.2 Discrétisation spatiale et résolution du problème	17
Conclusion	18
Annexes	19
Bibliographie	22

Bilan Technique

Fiche de synthèse		Clément THIBAUT - 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 trili-néaire HPC dans le code d'analyse de données Antares- Évaluer l'influence de la méthode d'interpola-tion 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— ...			
Résultats		Explications des écarts possibles	
<ul style="list-style-type: none">- Rapidité du code d'interpolation augmentée (quelle que soit la méthode), environ par 100 sur le cas test d'aéroacoustique- Méthode linéaire pour tous types de maillages rencontrés au CERFACS implémentée- Méthode linéaire généralement plus efficace que la méthode IDW dans les cas d'aéroacoustique		<ul style="list-style-type: none">— ...— ...	
Difficultés rencontrées		Travaux à poursuivre	
<ul style="list-style-type: none">- Prise en main des outils relativement fatigante au début- Échec de l'implémentation du 'multi-zones' avec points communs entre des zones pour l'interpolation linéaire		<ul style="list-style-type: none">- Implémenter une méthode d'ordre supérieur dans le code d'Antares	

Introduction

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 et Nadir, j'ai eu l'occasion de découvrir ce laboratoire et d'y candidater pour mon stage de M1. Carlos m'a trouvé un sujet sur l'interpolation dans le cas de post-processing de simulations numériques et son application en aéroacoustique. Le sujet m'a directement plu, 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 High Performance Computing (HPC), financé par sept actionnaires.

Antares[1] est une librairie python privée qui a été développée au CERFACS en 2012 et a pour objectif de faire du pré et post-processing sur des simulations numériques utilisées par les actionnaires du CERFACS.

Elle contient notamment une fonction d'interpolation, codée en Python.

Mon maître de stage, Carlos, est responsable d'Antares depuis dix mois.

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 ses contraintes associées).
- D'identifier les meilleurs paramètres pour l'équation Inverse Distance Weighting (IDW)
- D'implémenter la méthode trilinéaire que j'appellerais aussi barycentrique.

Pour bien comprendre, ce que nous voulons interpoler, ce sont les valeurs aux points d'un maillage dit 'target' grâce aux valeurs aux points d'un maillage 'source'. Par exemple dans le cadre d'un raffinement de maillage entre 2 itérations de calcul ou dans le cas de la création d'une sphère dans un maillage 3D pour l'application des équations de FWH¹ dans le cadre de la propagation aéroacoustique.

Pour expliquer plus en détails ce stage au CERFACS, je présenterai d'abord ce laboratoire, puis je vous exposerai le travail que j'ai réalisé.

1. FWH : Ffowcs Williams-Hawkings

Chapitre 1

Le CERFACS

Le Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique est un laboratoire de recherche privé avec pour actionnaires Airbus, le CNES (Centre d'Études Spatiales), EDF, Météo France, l'ONERA (Office National d'Études et de Recherche Aéronautiques), Safran et TotalEnergies. Il a pour but de développer la simulation numérique par le calcul haute performance (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 GIP (Groupement d'intérêt Public), pour devenir une société civile en 1996 et depuis 2021, le CERFACS est une SAS (Société par Actions Simplifiées).

Les deux bâtiments sont situés au 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 Scientistes y travaillent dans quatre équipes :

- Algo-Coop (Algorithmes Parallèles & sCientifics sOftware Operational Performances)
- CSG (Équipe Informatique et Support Utilisateur)
- ES (Energy et Safety)
- AAM (Advanced Aerodynamics and Multiphysics)
- GLOBC (Modélisation du climat et de son changement global)

La 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 à l'avance la tenue structurelle, les forces aérodynamiques, le volume sonore, etc. sans faire d'onéreux tests. L'équipe CFD (Computational Fluid Dynamics) est la plus grande du CERFACS. Elle se focalise sur la simulation des écoulements et de la combustion en développant des méthodes numériques avancées et en les appliquant aux avions, fusées, hélicoptères, moteurs, turbomachines, etc. Les liens de l'équipe CFD avec les autres équipes du CERFACS comme GLOBC ou PAE sont forts, car ces équipes utilisent aussi la CFD de façon intensive pour prévoir le changement climatique ou l'effet de l'aviation sur l'environnement : en effet, derrière ces thèmes, on retrouve en premier les équations qui régissent les écoulements des fluides.

Le CERFACS héberge des supercalculateurs (Kraken et Calypso) et un serveur sur lequel se trouve l'intranet avec toutes les ressources nécessaires aux employés. Nous y trouvons notamment un lien vers une page QVT¹...Et un lien vers la page du CSE²...avec énormément de ressources. Le CSSCT³ et le DUERP⁴ devrait obligatoirement y être présent selon le site du gouvernement. Je n'ai par réussi à trouver le document relatif à la RSE⁵. Il existe un document "Plan du management de la qualité"

1. QVT : Qualité de Vie au Travail

2. CSE : Comité Social et Économique

3. CSSCT : Commissions santé, sécurité et conditions de travail

4. DUERP : Document Unique d'Évaluation des Risques Professionnels

5. RSE : Responsabilité Sociétale des Entreprises

La question de l'environnement est très présente au CERFACS. Les employés sont poussés à venir à vélo. Ils donnent la prime (obligatoire pour les entreprises) pour chaque kilomètre fait à vélo sur le trajet habitation-travail-et vis-versa. Il y a un atelier de réparation et des emplacements sécurisés pour les vélos.

... Ils ont par exemple lancé 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.

Le cadre de travail m'a relativement plus, être assis dans un bureau climatisé avec une vue sur la campagne n'est pas désagréable. Un des seuls risques à ce type de travail est une mauvaise position qui peut entraîner des problèmes de dos, épaules, ... Pour palier cela il y a des affiches dans chaque bureau indiquant la position à avoir et des exercices. Ces mêmes informations apparaissent dans le 'livret d'accueil stagiaire' qui m'a été donné le premier jour. Il y a une cantine sur le site à quelques minutes à pied, où la plupart des chercheurs et thésards vont manger les midis.

Chapitre 2

Présentation du stage

Le stage a débuté au début du mois de juin 2024. Un ordinateur et un PIN-pad générateur d'OTP (One-Time-Password) ont été mis à ma disposition. Carlos, mon maître de stage, m'a fourni les instructions nécessaires pour comprendre et utiliser Antares[1], une librairie de développement, à l'aide de la documentation en ligne¹.

2.1 La librairie Antares

La structure des solutions CFD interprétées dans Antares est illustrée ci-dessous :

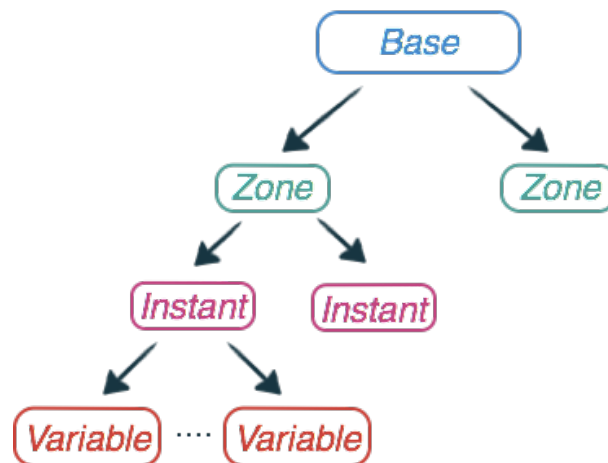


FIGURE 2.1 – Structure des données

Concrètement, le maillage peut être divisé en plusieurs zones. Ensuite chaque zone a un ou plusieurs instants où nous pouvons trouver la solution d'une variable, qui sera de même dimension que les coordonnées de l'instant.

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()
6 print(result[0][0]['v1'])
```

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

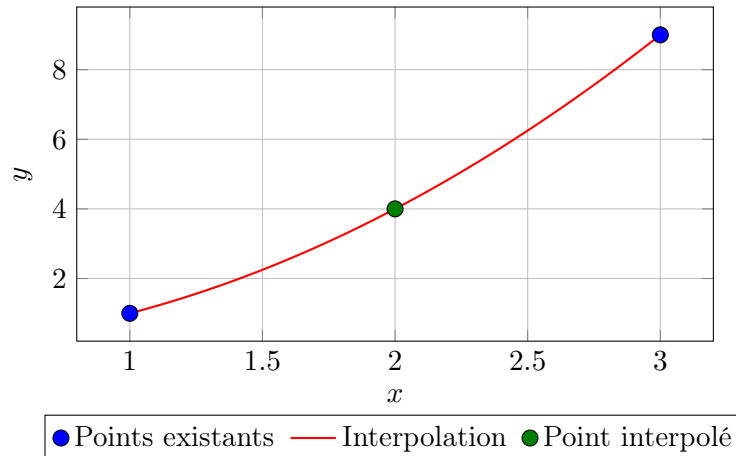
Un point important pour l'interpolation linéaire est la connectivité. Si le maillage est non structuré, alors, chaque cellule est définie par un type de forme et par ses sommets. Elle peut être donnée par `base[0][0]['tri']` si la base contient des triangles dans sa zone 0 à l'instant 0.

1. Documentation d'Antares : <https://cerfacs.fr/antares/>

Ensuite, la tâche consistait à résoudre un bug mineur sur Antares, ce qui a permis de se familiariser avec Nitrox, le Gitlab hébergé sur le serveur du CERFACS où sont situés Antares et d'autres codes du CERFACS.

2.2 Les différentes méthodes d'interpolation

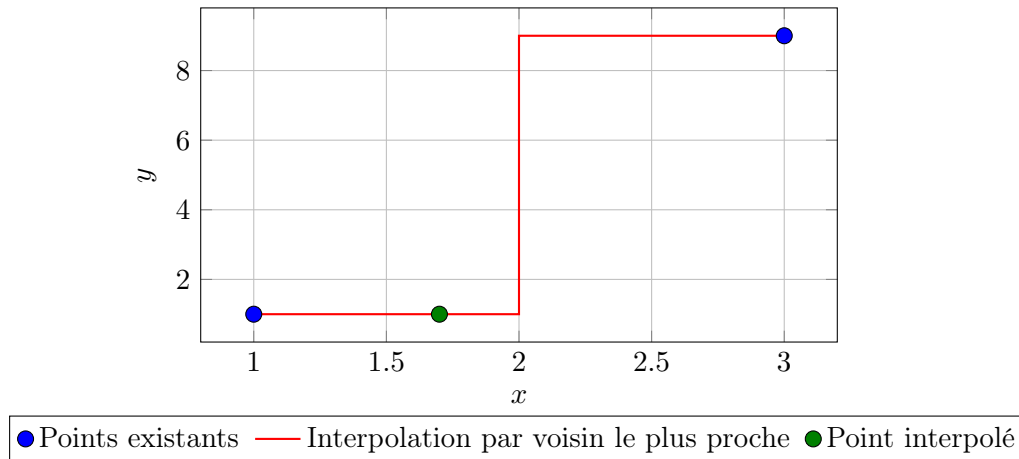
L'interpolation consiste à déterminer la valeur de nouveaux points à partir de la valeur de points existants. En voici un exemple en une dimension (l'axe x représente la position et l'axe y la valeur des points).



Nous allons premièrement présenter les types d'interpolation implémentables dans Antares, à savoir, qui permette de l'interpolation 3D, sur des maillages dits non structurés, c'est-à-dire pas de simples maillages, rectangulaires en 2D et hexaédrique en 3D, représentés par des matrices, mais des maillages créés avec différentes formes géométriques. Le temps de calcul, appelé 'coût' est aussi un paramètre à prendre en compte. Finalement, les caractéristiques mathématiques des équations à interpoler est probablement le paramètre le plus important à prendre en compte, mais aussi assurément le plus difficile. Effectivement différentes équations très difficiles à caractériser mathématiquement tel que l'équation de Naviers-Stokes sont utilisées et le niveau en maths est trop élevé pour pouvoir se plonger en profondeur dans ce problème. C'est pour cela qu'il n'y aura pas de résultat mathématique à présenter dans cette partie. Mais heureusement que ces méthodes ont déjà été implémentés et testés pour d'autres codes de simulation numérique, ce qui donne une bonne idée des résultats que nous pouvons espérer.

2.2.1 L'interpolation par voisin le plus proche

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



Cette méthode est discontinue et peu précise pour la plupart des fonctions.

2.2.2 L'interpolation 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 a pour formule :

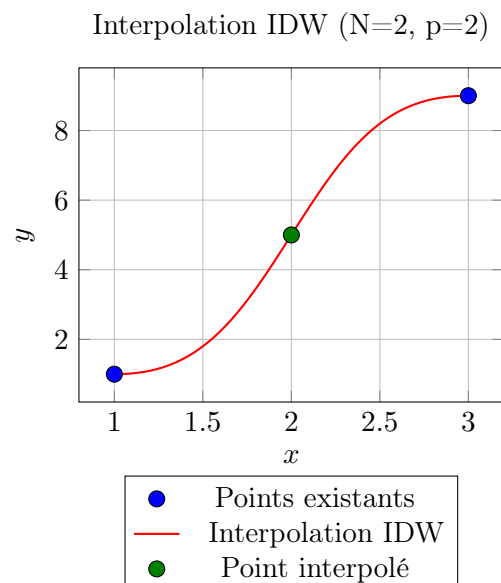
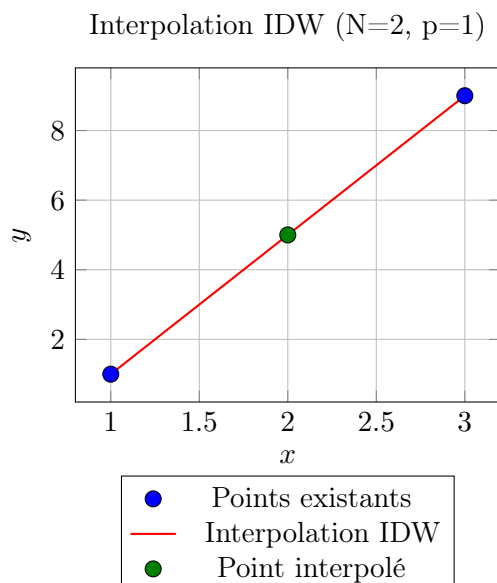
$$\hat{f}(x) = \frac{\sum_{i=1}^N \frac{f(x_i)}{d(x, x_i)^p}}{\sum_{i=1}^N \frac{1}{d(x, x_i)^p}}$$

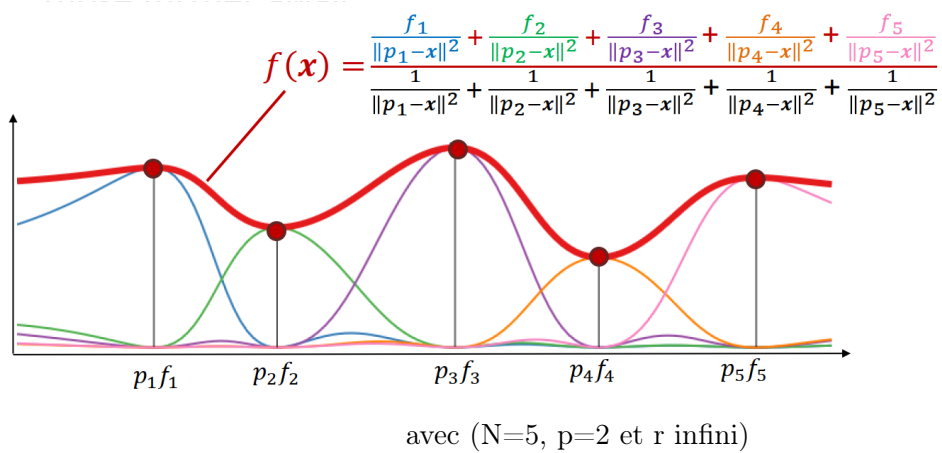
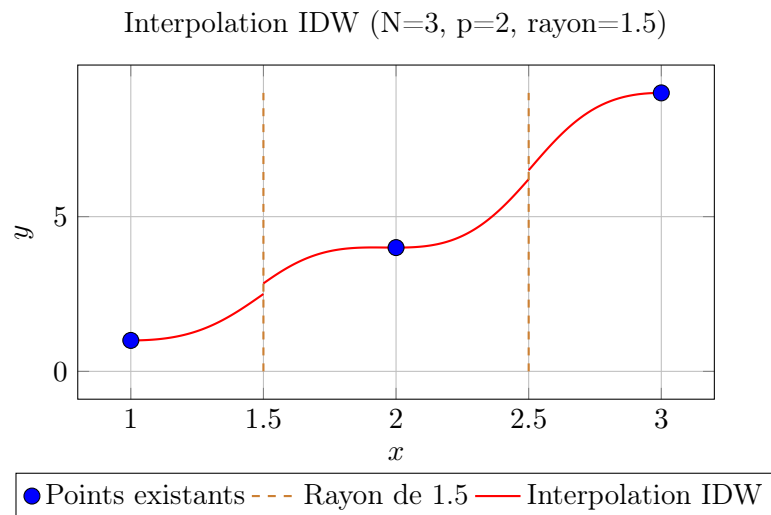
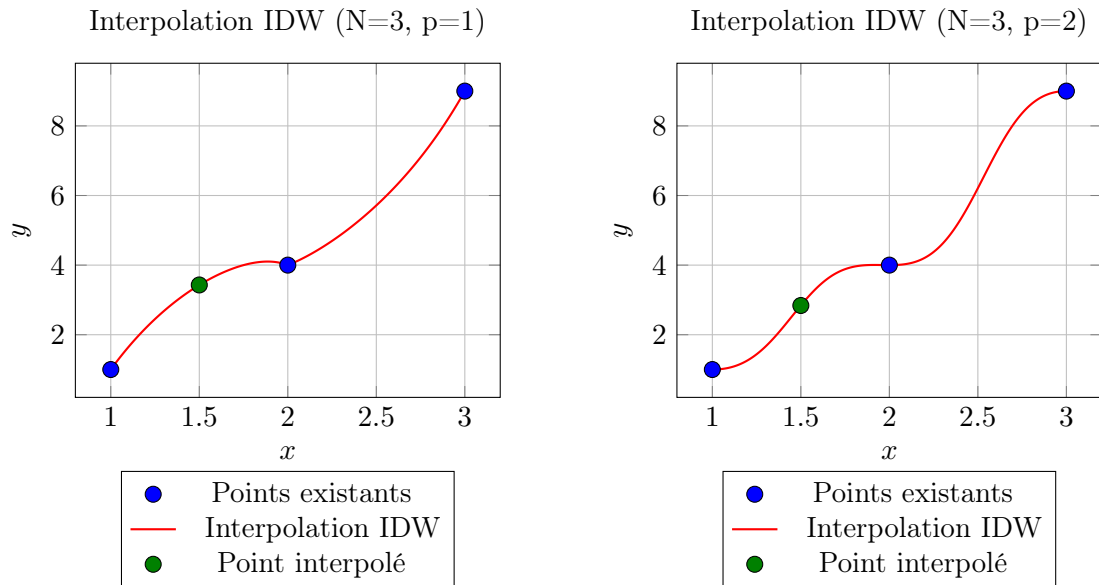
où :

- $\hat{f}(x)$ 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,
- N est le nombre total de points de données.

Pour éviter une division par 0, très proche d'un point existant, le point interpolé prendra la valeur du point existant.

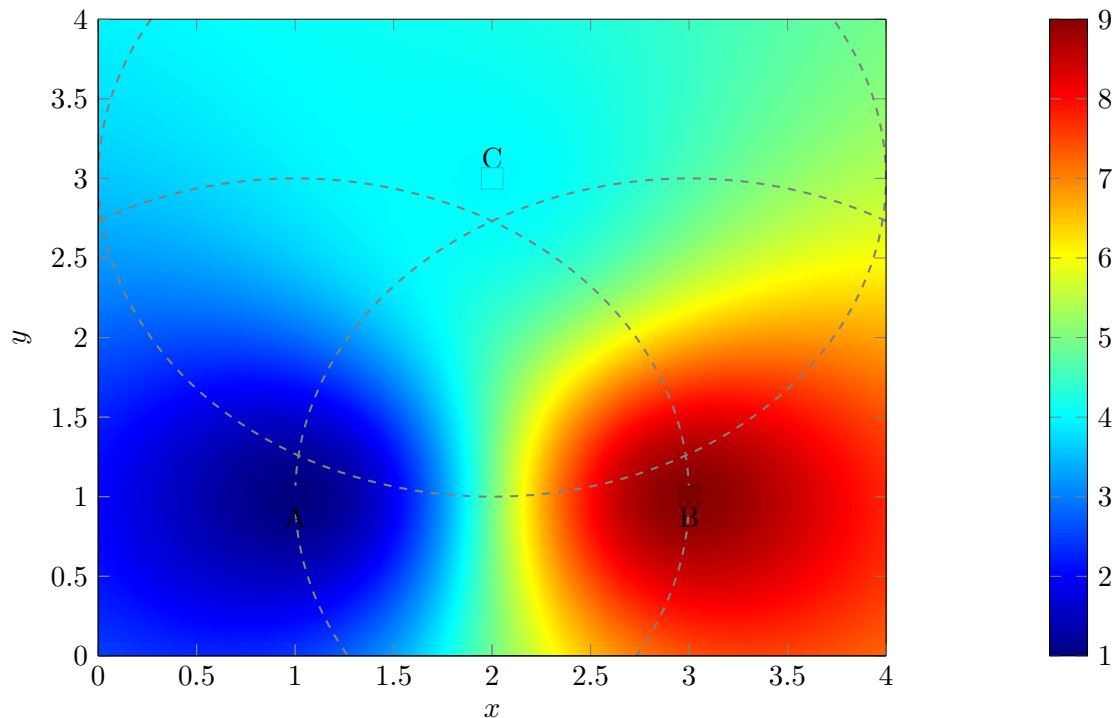
En voici une illustration :





Interpolation 1D

Interpolation IDW en 2D ($N=3$, $p=2$, Rayon=2.0 *non pris en compte dans le coloris*)



Nous comprenons assez vite que, si nous définissons une distance à partir de laquelle nous prenons en compte ou non un point, cette méthode est discontinue. Nous pouvons aussi imaginer que 2 points à droite et à gauche d'une ligne horizontale de 3 points (les plus proches), prendrons la même valeur pour $N = 3$, Schématiser.

D'autres méthodes dérivées ou similaires existent pour évaluer nos poids. La plus intéressante serait celle dite de Franke-Littke. Elle consiste à utiliser une distance maximale autour du point au-delà des autres points ne sont pas pris en compte. Autrement dit, en utilisant 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 car, confronté à un maillage ayant une différence de raffinement intrinsèque importante, dans certains cas aucuns points ne seraient pris, et dans d'autres, une grande somme serait calculée.

Nous pouvons modifier 2 paramètres : p et N . Par défaut, dans le code, $p = 1$ et N est égale aux nombres de sommets de la première cellule de la liste de types de cellules de la base target. Il y aurait peut-être une modification mineure à faire sur ce point. La structure de la base target importe peu à côté de celle de la base source, pour l'interpolation. Mais aucune modification qui pourrait changer l'utilisation qui a déjà été faite de ce traitement ne pourra être faite par soucis de rétrocompatibilité. ! Tests à faire ici !

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.

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 et thèses que j'ai lues. C'est pour cela que je présenterais plus tard comment j'ai trouvé des paramètres optimaux en faisant des tests.

2.2.3 L'interpolation polynomiale

2.2.4 L'interpolation par Splines

2.2.5 Méthodes géostatistiques

2.2.6 Méthode par moindres carrés

2.2.7 MISCOCG

2.2.8 L'interpolation linéaire

L'interpolation linéaire, est la plus simple (après les plus proches voisins et IDW). Elle est moyennement précise et ne demande que peu de ressource. Elle interpole généralement mieux qu'IDW. 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 (dans via d'autres codes qu'Antares). Cela les arrange donc aussi 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 points. Supposons que nous voulons interpoler une valeur d'un point p entre deux points a et b dans un espace 1D et que nous représentons leurs valeurs dans une deuxième dimension y . Nous aurons alors pour formule :

$$y_p = \frac{x_b - x_p}{x_b - x_a} \cdot y_a + \frac{x_p - x_a}{x_b - x_a} \cdot y_b$$

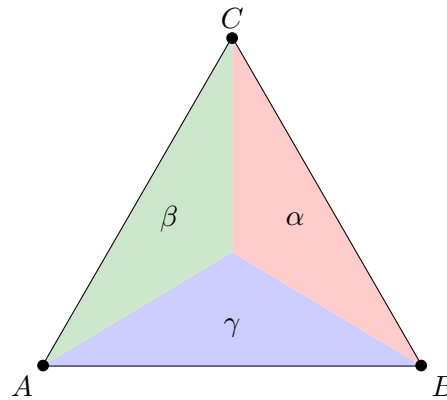
où y_p représente la valeur interpolée à la position x_p , et (x_a, y_a) et (x_b, y_b) sont les points de référence. J'ai écrit cette formule afin qu'elle soit symétrique par rapport aux points a et b , pour qu'ils jouent la même rôle. Ainsi elle s'entendra plus intuitivement dans des dimensions supérieures.

$\frac{x_b - x_p}{x_b - x_a}$ 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 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 contenus au niveau des nœuds. Aussi, Antares ne traite que des maillages ayant des valeurs uniquement au niveau des nœuds des cellules (pas entre). Il existe 3 principaux types de cellules (formes) en 2D : les triangles, les rectangles des maillages structurés (dans ce cas, la méthode est dite bilinéaire) et les quadrilatères (non croisés).

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 opposés et pondéré le tout par la surface du triangle.



Interpolation Barycentrique

Le calcul est $\frac{y_A \times \alpha + y_B \times \beta + y_C \times \gamma}{\alpha + \beta + \gamma}$.

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. La formule est l'extension de celle pour les triangles :

ÉQUATION

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. Cela correspond à deux interpolations linéaires. Souvent nous trouvons une équation analytique où tous les sommets ne jouent pas le même rôle, mais je trouvais cela plus simple de faire un calcul de poids pour pouvoir ensuite faire une moyenne pondérée :

Interpolation Linéaire sur un Rectangle

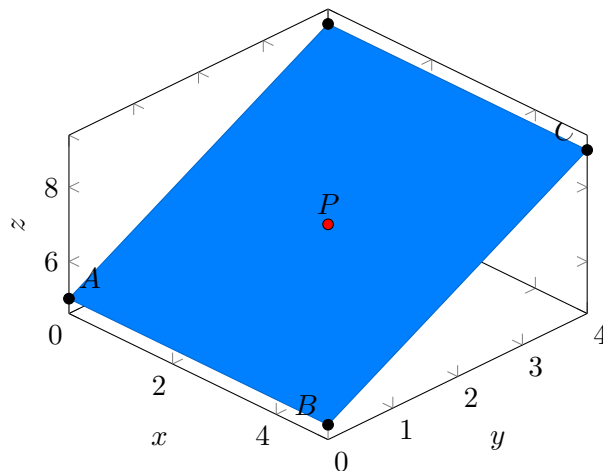


ILLUSTRATION à améliorer

Quadrilatère Viens maintenant la dernière forme 2D rencontrée dans les solutions traitées par Antares : les quadrilatères. Pour cela je n'ai pas trouvé de méthode. 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 forme 2D a comme valeur la moyenne de ses côtés. Via cette démarche, j'ai imaginé, graphiquement, tracer des traits entre le point à interpoler et les sommets de la forme dans laquelle il se situe (tel que pour l'interpolation Barycentrique). Cela permet de ne créer uniquement 4 sous formes. Ensuite pour déterminer le poids associé au sommet s_1 , il faut multiplier les deux surfaces qui lui sont opposés entre elles, et bien entendu, le pondérer une fois les autres poids calculés. Par opposé j'entends que ces surfaces ne sont composées d'aucune arête ayant pour l'une de leurs extrémités le point d'interpolation. Ceci est important pour le

3D. Pour l'instant, je n'ai démontré que par l'expérimentation que cette méthode était linéaire. Un point qui me perturbait était de faire des multiplications de surfaces, donc ordre 4, dans une méthode linéaire. Mais contrairement à son nom, l'interpolation bilinéaire est en réalité quadratique avec un résultat linéaire. On pourrait imaginer que, par chance, ma méthode soit quadratique. Premièrement j'ai vérifié et ce n'est apparemment pas le cas. Deuxièmement je pense que le quadratique n'englobe pas le linéaire dans le cas où nous nous basons uniquement sur les quatre points d'un quadrilatère. Effectivement, en 1D, si nous avons $f(x_i) = 0$ et $f(x_{i+1}) = 1$, le résultat d'une variable linéaire serait 0,5 et celui d'une variable quadratique 0,25, si nous avons uniquement connaissance de ces deux points. Normalement il faut s'appuyer sur plus de points pour le quadratique. Finalement voici l'équation :

ÉQUATION

ILLUSTRATION

3D

Pavé droit : Trilinéaire Pour le 3D, si le maillage est structuré, alors la forme 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. Nous associons comme poids à un des huit sommets s_1 le volume opposé, construit de la sorte :

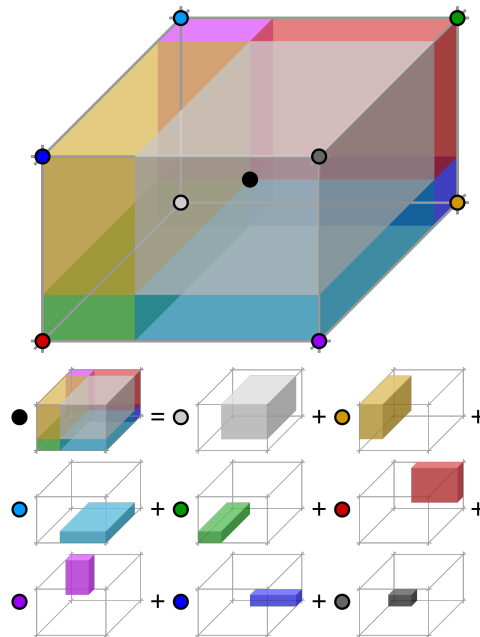


FIGURE 2.2 – Interpolation trilineaire

L'équation qui en découle est la suivante :

$$f(x, y, z) = \sum_{i=0}^1 \sum_{j=0}^1 \sum_{k=0}^1 f_{ijk} (1 - |x - x_i|)(1 - |y - y_j|)(1 - |z - z_k|) \quad (2.1)$$

2.2.9 Résumé des similitudes et différences des différentes méthodes

2.3 Implémentation de la méthode trilinéaire

2.3.1 La structure générale du code TreatmentInterpolation

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 target et renvoie dans le cas le plus simple la base target avec les valeurs interpolées. De manière simplifiée, dans le code, les zones de la base source sont fusionnées puis nous parcourons les instants. Cette fusion permet de faire un KDTree (pour arbre à k-dimensions), 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 target (et les distances associées, utilisés dans la méthode 'idw').

Ensuite nous calculons l'interpolation via une méthode 'principale' (`__idw_interpolate_instant` ou `__barycentrique_interpolate_instant`) qui peut elle-même appeler des fonctions ou méthodes.

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 appelée `node_to_elements`
- 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 l'élément
 - ◊ Si le point est à l'intérieur de la forme de l'élément :
 - 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 nan

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 source :

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

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

restent inchangés. Par exemple dans le cas de l'aéroacoustique, la solution contiens 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 dits 'partagés'. Si oui, alors les poids et autres données nécessaires sont calculés (grace à la variable `computation`), mais pas le premier instant. Ensuite tous les instants seront calculés avec les données. - Si le calcul sur la base source est réalisé plusieurs fois sans que

sa structure ne soit changée. Ceci arrive régulièrement, notamment lors de tests. Dans ce cas, il faut légèrement changer l'appel à la fonction, voici un exemple :

```
1  import antares
2  from copy import deepcopy
3  # deepcopy is used to be sure that the target_base will not be updated because
4    of the Treatment
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' can be stored on the computer for example, then you can call it back
13
14 myt = antares.Treatment('interpolation')
15 myt['source'] = source_base
16 myt['target'] = target_base
17 myt['data'] = data
18 result2 = myt.execute()
```

Listing 2.2 – 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 problème de rétrocompatibilité, car les résultats sont inchangés.

La méthode linéaire était initialement 200 à 500 fois plus lente que IDW. J'ai donc fait du 'profiling' pour savoir quelles lignes de code étaient les plus coûteuses. J'ai remplacé des listes par des arrays, ce qui augmente la rapidité d'accès aux éléments de ce dernier en mémoire. J'ai aussi optimisé le test de localisation d'un point dans une cellule et de calcul des poids. Je calcule d'abord les surfaces créées par le point cible avec les sommets puis, je calcule le volume de la cellule. Si ces volumes sont à peu près égaux, alors le point est dans la cellule et j'ai les poids. Sinon je sais qu'il faut chercher autre part.

2.3.4 Le résultat

2.4 Tests sur des cas d'aéroacoustique

Carlos a développé l'outil permettant de déterminer le résultat acoustique, à grande distance, à partir d'une surface, en utilisant les équations de Ffowcs Williams – Hawkins. Le résultat acoustique sont les petites variations de pression, impliquant du son (à différentes fréquences et amplitudes). En pratique, pour les utilisateurs d'Antares, cette surface est définie dans un maillage 'solution' où nous avons le résultat de la pression en différents points et différents instants.

2.4.1 Tests sur les paramètres de la méthode IDW

... (<https://cerfacs.fr/antares/>) :
— TreeMesh

2.4.2 Discrétisation spatiale et résolution du problème

Conclusion

Annexes

Liste des acronymes

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

HPC High Performance Computing

IDW Inverse Distance Weighting

Bibliographie

- [1] ANTARES DEVELOPMENT TEAM. *Antares Documentation Release 2.2.0*. Avr. 2024. URL : <https://cerfacs.fr/antares/>.
- [2] Chandrajit L BAJAJ. *Multi-dimensional Hermite Interpolation and Approximation for Modelling and Visualization*.
- [3] J A BENEK et al. *Chimera : A Grid-Embedding Technique* ARNOLD ENGINEERING DEVELOPMENT CENTER ARNOLD AIR FORCE STATION, TENNESSEE AIR FORCE SYSTEMS COMMAND UNITED STATES AIR FORCE.
- [4] C BENOIT, G JEANFAIVRE et E CANONNE. *Synthesis of Onera Chimera Method Developed in the Frame of Chance Program*. 2023.
- [5] C BENOIT et al. *Cassiopee : a CFD pre-and post-processing tool*. 2023. URL : <https://hal.science/hal-01141585>.
- [6] Ricardo CAMARERO. *MEC6212 : Génération de maillages MAILLAGES TRANSFINIS*. 2023.
- [7] William M CHAN. *Development of Numerical Methods for Overset Grids with Applications for the Integrated Space Shuttle Vehicle*. 1995.
- [8] G. CUNHA et S. REDONNET. « Development of optimized interpolation schemes with spurious modes minimization ». In : *International Journal for Numerical Methods in Fluids* 80 (2 jan. 2016), p. 140-158. ISSN : 10970363. DOI : 10.1002/flid.4079.
- [9] Guilherme CUNHA et Stephane REDONNET. *An Innovative Interpolation Technique for Aeroacoustic Hybrid Methods*. 2011.
- [10] FL FLUIDS ENGINEER, FL MARCELLILIE et FL PAULLSCHALLHORN. *Interpolation Method Needed for Numerical Uncertainty Analysis of Computational Fluid Dynamics*.
- [11] William J GORDON et Charles A HALL. « CONSTRUCTION OF CURVILINEAR COORDINATE SYSTEMS AND APPLICATIONS TO MESH GENERATION ». In : *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING* 7 (1973), p. 461-477.
- [12] William J GORDON. *BLENDING-FUNCTION METHODS OF BIVARIATE AND MULTIVARIATE INTERPOLATION AND APPROXIMATION**. 1971. URL : <http://www.siam.org/journals/ojsa.php>.
- [13] *Internal Documentation*. 2023.
- [14] Soohyeon KIM, Sooahm RHEE et Taejung KIM. « Digital surface model interpolation based on 3D mesh models ». In : *Remote Sensing* 11 (jan. 2019). ISSN : 2072-4292. DOI : 10.3390/rs11010024.
- [15] 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* 363 (juin 2018), p. 371-398. ISSN : 10902716. DOI : 10.1016/j.jcp.2018.02.045.
- [16] David LEVIN. *Mesh-Independent Surface Interpolation*.
- [17] Jens-Dominik MÜLLER. *A user's guide to h i p*. 2020.

- [18] Taku OZAWA et Takahiko TANAHASHI. *CIVA (Cubic Interpolation with Volume/Area Coordinates) and AMR (Adaptive Mesh Refinement) Method for Discrete Boltzmann Equation* *.
- [19] Grant E PALMER. *Construction of CFD Solutions Using Interpolation Rather than Computation with the ADSI Code*. 2009.
- [20] Wiebke K PEITSCH et al. *Drebrin, an Actin-Binding, Cell-Type Characteristic Protein : Induction and Localization in Epithelial Skin Tumors and Cultured Keratinocytes*. 2005.
- [21] Stéphanie PÉRON et Christophe BENOIT. « Automatic off-body overset adaptive Cartesian mesh method based on an octree approach ». In : *Journal of Computational Physics* 232 (jan. 2013), p. 153-173. ISSN : 10902716. DOI : 10.1016/j.jcp.2012.07.029.
- [22] Alain PERRONNET. *Interpolation transfinie SW le triangle, le tetraèdre et le pentakdre. Application h la crhation de maillages et h la condition de Dirichlet*. 1998. URL : %5Curl%7Bhttp://www.ann.jUssirrl.fr/~rro~n~%7D.
- [23] S REDONNET. « Issue 7-June 2014-Aircraft Noise Prediction via Aeroacoustic Hybrid Methods : Development and Application of Onera Tools over the Last Decade : Some Examples ». In : (). DOI : 10.12762/2014.AL07-07.
- [24] Stefan J. SCHODER et al. « Conservative source term interpolation for hybrid aeroacoustic computations ». In : *25th AIAA/CEAS Aeroacoustics Conference, 2019*. American Institute of Aeronautics et Astronautics Inc, AIAA, 2019. ISBN : 9781624105883. DOI : 10.2514/6.2019-2538.
- [25] Norman E SUHS, Stuart E ROGERS et William E DIETZ. *PEGASUS 5 : An Automated Pre-Processor for Overset-Grid CFD*. 2023.
- [26] Nobuatsu TANAKA, Toshiteru YAMASAKI et Takaya TAGUCHI. *Accurate and Robust Fluid Analysis Using Cubic Interpolation with Volume/Area Coordinates (CIVA) Method on Unstructured Grids* *.
- [27] Manel TAYACHI. *Couplage de modèles de dimensions hétérogènes et application en hydrodynamique*. 2013. URL : <https://theses.hal.science/tel-00930084>.
- [28] Paul WALKER, Ulrich KROHN et David CARTY. *ARBTools : A tricubic spline interpolator for three-dimensional scalar or vector fields*. 2019. DOI : 10.5334/jors.258.

Résumé en français.
Puis en anglais.
Attention, doit se trouver sur une page de numéro paire.