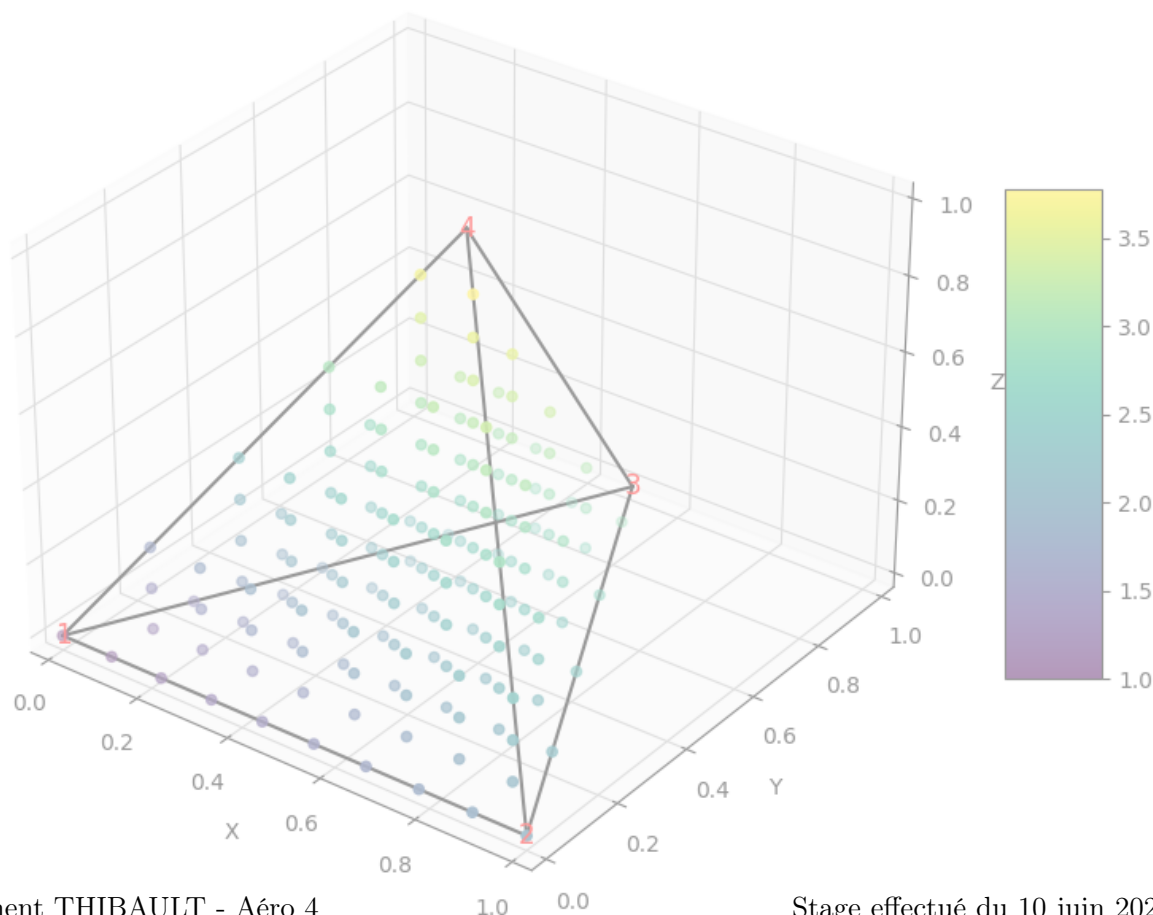


## Rapport de stage Aéro 4

# Développement d'une méthode d'interpolation trilinéaire et évaluation de ses performances dans l'application de l'aéroacoustique

Stagiaire au sein de l'équipe CFD

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<sup>1</sup> 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 explication 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 apporté au CERFACS.

# Table des matières

<b>Bilan Technique</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>1 Le CERFACS</b>	<b>6</b>
<b>2 Présentation du stage</b>	<b>7</b>
2.1 Mission 1 : Recherche les différentes méthodes d'interpolation . . . . .	7
2.1.1 L'interpolation par voisin le plus proche . . . . .	8
2.1.2 L'interpolation IDW . . . . .	8
2.1.3 L'interpolation polynomiale . . . . .	8
2.1.4 L'interpolation par Splines . . . . .	8
2.1.5 Méthodes géostatiques . . . . .	8
2.1.6 Méthode par moindres carré . . . . .	8
2.1.7 MISCOG . . . . .	8
2.1.8 L'interpolation linéaire . . . . .	8
2.1.9 Résumé des similitudes et différences des différentes méthodes . . . . .	10
2.2 Mission 2 : Implémenter la méthode trilinéaire . . . . .	10
2.2.1 La structure générale du code TreatmentInterpolation . . . . .	10
2.2.2 L'algorithme . . . . .	11
2.2.3 ... . . . .	11
2.2.4 Les difficultés . . . . .	11
2.2.5 Le résultat . . . . .	11
2.3 Mission 3 : Tester sur des cas d'aéroacoustique . . . . .	11
2.3.1 Tests sur les paramètres de la méthode IDW . . . . .	11
2.3.2 Discrétisation spatiale et résolution du problème . . . . .	11
<b>Conclusion</b>	<b>12</b>
<b>Annexes</b>	<b>13</b>
<b>Bibliographie</b>	<b>14</b>

## Bilan Technique

Fiche de synthèse		Clément THIBAUT - Aéro 4	
Sujet de stage		Objectifs	
<p>Influence de la méthode d'interpolation sur la propagation acoustique FWH <sup>a</sup></p> <ul style="list-style-type: none"><li>— Développer une méthode d'interpolation trilinéaire HPC dans le code d'analyse de données Antares</li><li>— Évaluer l'influence de la méthode d'interpolation dans la qualité des résultats de propagation acoustique avec l'analogie Ffowcs Williams – Hawkings</li><li>— Améliorer les performances HPC de la méthode d'interpolation dans Antares</li></ul> <hr/> <p><sup>a</sup>. FWH :..</p>		<ul style="list-style-type: none"><li>— Développer une méthode d'interpolation trilinéaire HPC dans le code d'analyse de données Antares</li><li>— Évaluer l'influence de la méthode d'interpolation dans la qualité des résultats de propagation acoustique avec l'analogie Ffowcs Williams – Hawkings</li><li>— Améliorer les performances HPC de la méthode d'interpolation dans Antares</li></ul>	
Client principal		Outils utilisés	
<ul style="list-style-type: none"><li>— CERFACS (AIRBUS, Safran)</li><li>— Date de mise à jour : octobre 2024</li></ul>		VSCode, Kraken (supercalculateur du CERFACS), Antares, Paraview	
Études réalisées			
<ul style="list-style-type: none"><li>— Influence des paramètres 'n' et 'p' de la méthode d'interpolation 'IDW'</li><li>— ...</li></ul>			
Résultats		Explications des écarts possibles	
<ul style="list-style-type: none"><li>— Rapidité du code d'interpolation augmentée (quelque soit la méthode), environ par 100 sur le cas test d'aéroacoustique</li><li>— Méthode trilinéaire pour tous types de maillages rencontrés au CERFACS implémentée</li><li>— Méthode trilinéaire généralement plus efficace que la méthode 'IDW' dans les cas d'aéroacoustique</li></ul>		<ul style="list-style-type: none"><li>— ...</li><li>— ...</li></ul>	
Difficultés rencontrées		Travaux à poursuivre	
<ul style="list-style-type: none"><li>— Prise en main des outils relativement fatigante au début</li><li>— Echec de l'implémentation du 'multi-zones' pour l'interpolation trilinéaire</li></ul>		<ul style="list-style-type: none"><li>— Implémenter une méthode d'ordre supérieur dans le code d'Antares</li></ul>	

## Introduction

J'adore les mathématiques appliqués, la mécanique des fluides et je voulais découvrir le monde de la recherche. Lors d'une présentation des activités au 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 pour 6 actionnaires qui utilisent des modèles numériques pour simuler des écoulements dans le monde réel.

Antares 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 X temps.

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).

- De trouver s'il existait des meilleurs paramètres  $N$  et  $p$  à l'équation déjà existante IDW (Inverse Distance Weighting).

- D'implémenter la méthode trilineaire 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 FWH (Ffowcs Williams – Hawkins) dans le cadre de la propagation aéroacoustique.

Pour expliquer plus en détails ce stage au CERFACS, je présenterais dans une première partie ce laboratoire (...) pour ensuite vous présenter le travail que j'ai réalisé.

# 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 Spatiale), Météo France, l'ONERA (Office National d'Études et de Recherche Aérospatiales), Safran et TotalEnergie. 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. (Je pourrais ici mettre la longue liste des 11 membres du début..). Il a été créé en 1988 sous le statut de GIP (Groupement d'Interêt 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 (Equipe Informatique et Support Utilisateur)
- CFD (Mécanique des fluides numérique)
- GLOBC (Modélisation du climat et de son changement global)

/! : bug titres base de page /!

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, 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 cadre de travail m'a relativement plus, être assis dans un bureau climatisé avec une vue sur la campagne n'est pas désagréable. Il y a une cantine sur le site à quelques minutes à pied, où la plupart des chercheurs et thésards vont manger tous les midi.

## Chapitre 2

# Présentation du stage

Mon stage a commencé le lundi 10 juin 2024. Après avoir récupéré un ordinateur et un PIN-pad générateur d'OTP (One-Time-Password), Carlos, mon maître de stage a pu me montrer comment comprendre la librairie que j'allais développer : Antares. Une ressource sur l'intranet ? permet d'apprendre à l'utiliser. Pour vous illustrer la base d'une solution CFD que nous pourrions vouloir traiter avec Antares, voici un schéma :

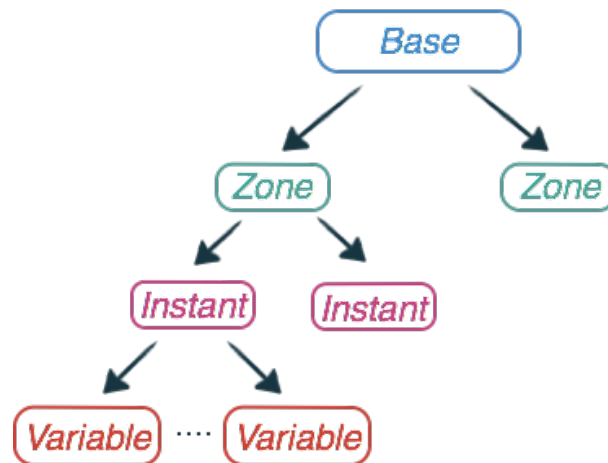


FIGURE 2.1 – Structure des données

Ensuite, j'ai été chargé de résoudre un petit bug sur Antares, ce qui m'a permis de prendre la main sur Nitrox, le gitlab hébergé sur le serveur du CERFACS où se situe Antares et d'autres codes du CERFACS.

### 2.1 Mission 1 : Recherche les différentes méthodes d'interpolation

Ma première mission a été de recenser les méthodes d'interpolation qui seraient implémentables dans Antares, à savoir, qui permettent de l'interpolation 3D, sur des maillages dits non structuré, 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 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 tel que l'équation de Naviers-Stokes sont utilisées et mon niveau en maths est trop limité pour pouvoir me plonger en profondeur dans ce problème. C'est pour cela que je n'ai 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.

### 2.1.1 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  $p$ . Nous comprenons assez vite que cette méthode est discontinue et même pas linéaire... 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 dérivés ou similaires existent pour évaluer nos poids. La plus intéressante serait celle dite de Franke-Littke. Elle consiste à utiliser une distance maximal autour du points au-delà les 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 utile 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.

### 2.1.2 L'interpolation IDW

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

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.

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 implémenté dans Antares.

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 forme de cellule de la liste de formes de cellules de la base target (je n'ai pas compris pourquoi pas de la base source) /! Tests à faire ici /! Ces informations ne sont pas confidentielles Carlos ?

Nous remarquons que pour  $n = 1$ , nous retrouvons la méthode du voisin le plus proche, pour tout  $p$ .

Une de mes mission était de chercher s'il y avait des paramètres plus optimisés que  $N$  et  $p$  pour cette méthode. Je n'ai pas trouvé le réponse dans les différents articles et thèses que j'ai lu. C'est pour cela que je présenterais plus tard comment j'ai trouvé des paramètre optimaux en faisant des tests.

### 2.1.3 L'interpolation polynomiale

### 2.1.4 L'interpolation par Splines

### 2.1.5 Méthodes géostatiques

### 2.1.6 Méthode par moindres carré

### 2.1.7 MISCOG

### 2.1.8 L'interpolation linéaire

Aussi appelée interpolation Barycentrique, l'interpolation linéaire, est la plus simple (après les plus proches voisins et IDW), et la plus utilisée par Airbus, Safran et d'autres industriels (dans via d'autres codes qu'Antares). C'est pour cela qu'ils ont demandé au CERFACS de l'implémenter des Antares, car ils l'utilisent actuellement via d'autres moyen. 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'il 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 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  $b$ .

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 maillages sont contenues 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 23 principales types de cellules (formes) en 2D : les triangles 'tri', et les quadrilatères 'qua' (non croisés) et les rectangles des maillages structurés. 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 au points pondéré par la surface opposé et pondéré le tout par la surface du triangle.

En ce qui concerne l'interpolation sur un rectangle, nous la trouvons aussi facilement. La formule est l'extension de celle pour les triangles :

#### EQUATION

Visuellement nous créons cette fois des traits parallèles au passant par le point d'interpolation et nous additionnons, de manière pondérée, les 4 surfaces multipliés chacune par leurs 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 :

#### ILLUSTRATION

Viens maintenant la dernière forme 2D rencontrée dans les solutions traités par Antares : les 'qua'. 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 cotés. 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 composés d'aucune arête ayant pour l'une de leurs extrémité 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

la 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. Finalement voici l'équation :

EQUATION

ILLUSTRATION

Pour le 3D, si le maillage est structuré, alors la forme est le pavé droit. A ce moment nous sommes dans le cas de l'interpolation dite trilinéaire. Encore une fois la formule se trouve facilement. Nous associons comme poid à un des huit sommet  $s_1$  le volume opposé, construit de la sorte :

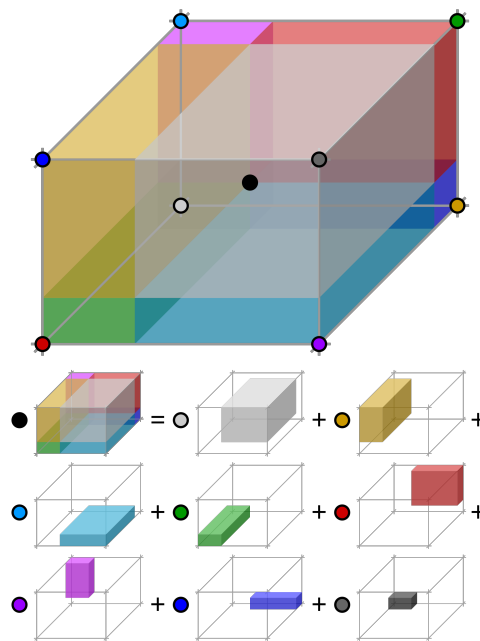


FIGURE 2.2 – Interpolation trilinéaire

L'équation qui en découle est la suivante :

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

### 2.1.9 Résumé des similitudes et différences des différentes méthodes

## 2.2 Mission 2 : Implémenter la méthode trilinéaire

### 2.2.1 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. Grace 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 interpolés. 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 associés, 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.

### 2.2.2 L'algorithme

Si vraiment ce n'est pas confidentiel je peux mettre le code en annexe Carlos.

Premièrement, dans `__barycentrique_interpolate_instant`, nous récupérons le nombre de points maximal sur lequel nous allons nous appuyer pour l'interpolation, soit le nombre maximale de sommet des formes présentes dans le maillage. Nous récupérons les distances et les indices du KDtree. Ensuite nous devons réarranger des indices qui se sont fait déplacer lors de la fusion des zones. Nous récupérons aussi différentes variables, comme les coordonnées des points, ...

Nous récupérons d'une fonction définie plus bas (`get_list_cell_type`) : (`list_cell_type`, `max_node_index`)

Dans le cas où le maillage est le même entre tous les instants, nous ne recalculons pas tout ces paramètres, ce qui permet une diminution significative du temps de calcul.

Expliquer 'is point on cell', etc...

### 2.2.3 ...

#### EXEMPLE DE MISE EN FORME PYTHON

```
1 def __idw_interpolate_instant(x, y, z, values, power=2):
2     # Calculer l'interpolation IDW
3     weights = [(1 / (distance**power)) for distance in distances]
4     interpolated_value = sum(w * v for w, v in zip(weights, values)) / sum(
5         weights)
6     return interpolated_value
7
8 def __barycentrique_interpolate_instant(x, y, z, values):
9     # Calculer l'interpolation barycentrique
10    weights = [barycentric_weight(x, y, z, vertex) for vertex in vertices]
11    interpolated_value = sum(w * v for w, v in zip(weights, values))
12    return interpolated_value
```

Listing 2.1 – Calcul de l'interpolation

### 2.2.4 Les difficultés

### 2.2.5 Le résultat

## 2.3 Mission 3 : Tester 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.

### 2.3.1 Tests sur les paramètres de la méthode IDW

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

### 2.3.2 Discrétisation spatiale et résolution du problème

# Conclusion

# Bibliographie

- [1] Chandrajit L BAJAJ. *Multi-dimensional Hermite Interpolation and Approximation for Modelling and Visualization*.
- [2] 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.
- [3] C BENOIT, G JEANFAIVRE et E CANONNE. *Synthesis of Onera Chimera Method Developed in the Frame of Chance Program*. 2023.
- [4] C BENOIT et al. *Cassiopee : a CFD pre-and post-processing tool*. 2023. URL : <https://hal.science/hal-01141585>.
- [5] Ricardo CAMARERO. *MEC6212 : Génération de maillages MAILLAGES TRANSFINIS*. 2023.
- [6] William M CHAN. *Development of Numerical Methods for Overset Grids with Applications for the Integrated Space Shuttle Vehicle*. 1995.
- [7] 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.
- [8] Guilherme CUNHA et Stephane REDONNET. *An Innovative Interpolation Technique for Aeroacoustic Hybrid Methods*. 2011.
- [9] FL FLUIDSSENGINEER, FL MARCELLILIE et FL PAULLSCHALLHORN. *Interpolation Method Needed for Numerical Uncertainty Analysis of Computational Fluid Dynamics*.
- [10] 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.
- [11] William J GORDON. *BLENDING-FUNCTION METHODS OF BIVARIATE AND MULTIVARIATE INTERPOLATION AND APPROXIMATION\**. 1971. URL : <http://www.siam.org/journals/ojsa.php>.
- [12] *Internal Documentation*. 2023.
- [13] 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.
- [14] 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.
- [15] David LEVIN. *Mesh-Independent Surface Interpolation*.
- [16] Jens-Dominik MÜLLER. *A user's guide to h i p*. 2020.
- [17] Taku OZAWA et Takahiko TANAHASHI. *CIVA (Cubic Interpolation with Volume/Area Coordinates) and AMR (Adaptive Mesh Refinement) Method for Discrete Boltzmann Equation\**.

- [18] Grant E PALMER. *Construction of CFD Solutions Using Interpolation Rather than Computation with the ADSI Code*. 2009.
- [19] Wiebke K PEITSCH et al. *Drebrin, an Actin-Binding, Cell-Type Characteristic Protein : Induction and Localization in Epithelial Skin Tumors and Cultured Keratinocytes*. 2005.
- [20] 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.
- [21] Alain PERRONNET. *Interpolation transfinie SW le triangle, le tétraèdre 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.
- [22] 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.
- [23] 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.
- [24] Norman E SUHS, Stuart E ROGERS et William E DIETZ. *PEGASUS 5 : An Automated Pre-Processor for Overset-Grid CFD*. 2023.
- [25] Nobuatsu TANAKA, Toshiteru YAMASAKI et Takaya TAGUCHI. *Accurate and Robust Fluid Analysis Using Cubic Interpolation with Volume/Area Coordinates (CIVA) Method on Unstructured Grids \**.
- [26] Manel TAYACHI. *Couplage de modèles de dimensions hétérogènes et application en hydrodynamique*. 2013. URL : <https://theses.hal.science/tel-00930084>.
- [27] 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.