

Rapport Technique

# Test de la fonctionnalité ordre élevé du coupleur de codes CWIPI et Intégration dans OpenPALM

Romain Casta – Thierry Morel

**TR-CMGC-19-95**

Juin 2019



## 1. INTRODUCTION

Le développement de logiciel de calculs basés sur des méthodes numériques d'ordre élevé compactes telles que les méthodes Galerkin discontinu ou spectrales est en plein essor. Ces méthodes utilisent des maillages d'ordre élevé. Les logiciels de maillage génèrent depuis peu des éléments d'ordre élevé courbes qui épousent au mieux la CAO des domaines maillés. La courbure locale d'un élément est définie par un polynôme dont l'ordre est celui de l'élément.

CWIPI est utilisé depuis une petite dizaine d'années pour coupler des logiciels basés sur des méthodes numériques Galerkin discontinu et volume fini essentiellement dans le domaine de l'aéroacoustique [1][2][3][4][6][7]. Cette utilisation ne peut s'effectuer que dans un contexte où les éléments géométriques sont linéaires donc droits. En outre, la mise en œuvre de ces couplages est complexe car il est nécessaire d'écrire un callback permettant de définir la méthode d'interpolation à partir de la base fonctionnelle de la méthode d'ordre élevé utilisée.

Les travaux réalisés depuis quelques années à l'ONERA permettent maintenant à CWIPI de prendre en compte nativement les éléments courbes d'ordre quelconque et proposent une méthode d'interpolation par défaut pour ce type d'élément. Ceci simplifie la mise en œuvre du couplage en évitant à l'utilisateur d'écrire la fonction callback d'interpolation. Ces travaux se limitent pour l'instant aux éléments surfaciques. Ils sont disponibles dans la version 0.10.3 de CWIPI téléchargeable à l'adresse <https://w3.onera.fr/cwipi/sites/w3.onera.fr/cwipi/files/u4/cwipi-0.10.3.tgz>.

La phase de spécification a dû répondre aux contraintes suivantes :

- Souplesse d'utilisation,
- Définition simple et générique d'un élément d'ordre quelconque,
- Possibilité de dissocier ordre géométrique et ordre de la méthode numérique,
- Pas de modification de l'interface d'appel des fonctions existantes,
- Compatibilité entre nouvelles et anciennes fonctions pour assurer des couplages entre une interface géométrique d'ordre élevée et une interface géométrique définie à l'aide des fonctions préexistantes,
- Pas de dégradation de la performance d'ensemble de CWIPI.

Nous avons contribué au CERFACS à son développement, en validant ces fonctionnalités par des tests de scalabilité forte jusqu'à 8000 coeurs et en intégrant ces nouvelles fonctionnalités dans OpenPalm. Un exemple de couplage combustion-acoustique avec les codes CEDRE/SPACE développée à l'ONERA est également présenté.

## 2. VALIDATION PAR ETUDE DE SCALABILITÉ

La validation de la nouvelle fonctionnalité s'est déroulée en deux temps. Une première étape a été consacrée à la validation de l'algorithme de calcul d'un point à un élément courbe. La seconde a permis de valider la fonctionnalité couplage surfacique d'ordre élevé dans sa globalité. C'est cette deuxième étape que nous avons effectuée au CERFACS via un cas test dédié décrit ci-dessous.

## 1. Géométrie Sphérique « fortran\_surf\_PIQJ »

Ce cas test permet de vérifier la bonne prise en compte des éléments d'ordre élevé courbes. La Figure 1 présente les deux couplages traités. Le premier effectue des échanges entre une sphère triangulée et une sphère quadrangulée composées d'éléments linéaires. Le second reprend les mêmes maillages mais composés d'éléments courbes d'ordre 2.

Le champ interpolé est le champ des coordonnées. L'erreur d'interpolation de ce champ représente la distance entre le point et son projeté sur le maillage source. Ceci permet de quantifier l'approximation géométrique de la sphère par chacun des quatre maillages. Les erreurs moyennes sont les suivantes :

- ✓ Maillage quadrangle d'ordre 1 :  $1,68 \cdot 10^{-02}$ ,
- ✓ Maillage triangle d'ordre 1 :  $1,38 \cdot 10^{-02}$ ,
- ✓ Maillage quadrangle d'ordre 2 :  $4,10 \cdot 10^{-04}$ ,
- ✓ Maillage triangle d'ordre 2 :  $2,95 \cdot 10^{-04}$

Ces résultats confirment que les maillages d'ordre 2 représentent mieux la surface de la sphère que les maillages d'ordre 1 comme on peut le constater visuellement sur la Figure 1.

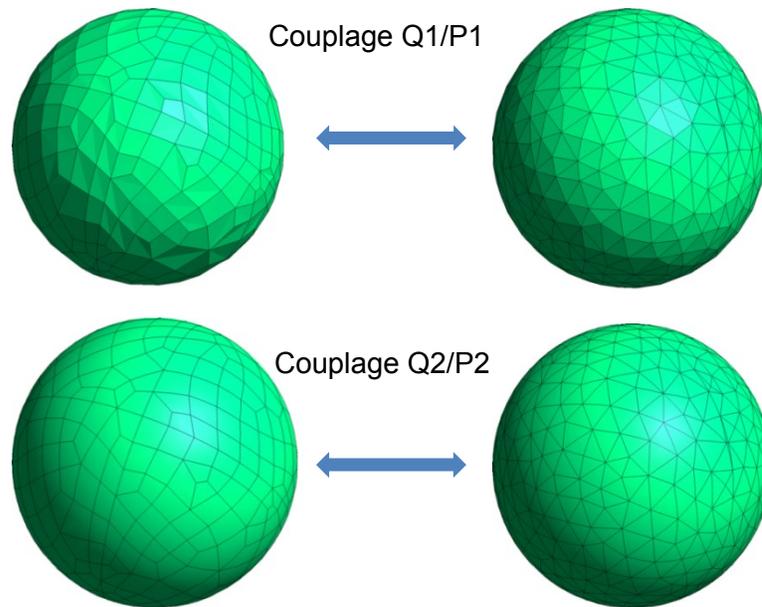


Figure 1 : Maillages du cas "fortran\_surf\_PIQJ"

## 2. Etude de scalabilité

L'étude de scalabilité forte a été réalisée sur la machine beaufix de Météo France. Les deux cas testés sont issus des deux cas de validation présentés ici. La taille des maillages a été fortement augmentée et pour des raisons de facilité de génération de maillage, toutes les surfaces sont composées uniquement de triangles. Les maillages ont été générés par le mailleur GMSH [8].

## 2.1 Surface plane

La surface d'échange plane est maillée de part et d'autre par 243 600 et 221 768 triangles. 15 millions de points sont localisés, interpolés et échangés à travers chacun des maillages.

La Figure 2 montre que quelque soit l'ordre, la courbe de scalabilité du temps de localisation conserve la même tendance. Le traitement des éléments d'ordre élevé ne dégrade pas les performances de la bibliothèque sur un nombre très important de cœurs de calcul.

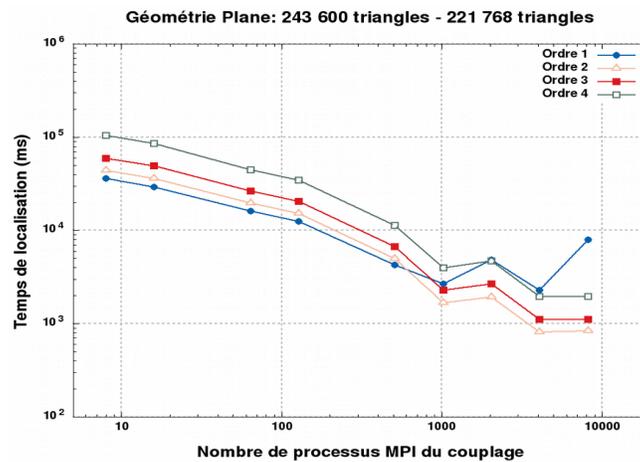


Figure 2 : Scalabilité sur maillage plan – Temps de localisation

Concernant le temps d'interpolation et d'échange du champ, la Figure 3 montre également que quelque soit l'ordre la courbe de scalabilité conserve la même tendance. Sur un très grand nombre de cœurs, ce temps devient quasiment identique quelque soit l'ordre. Ceci est probablement dû au fait que le temps d'échange devient prédominant par rapport au temps de calcul lié à l'interpolation en raison du peu de points à traiter sur chaque cœur.

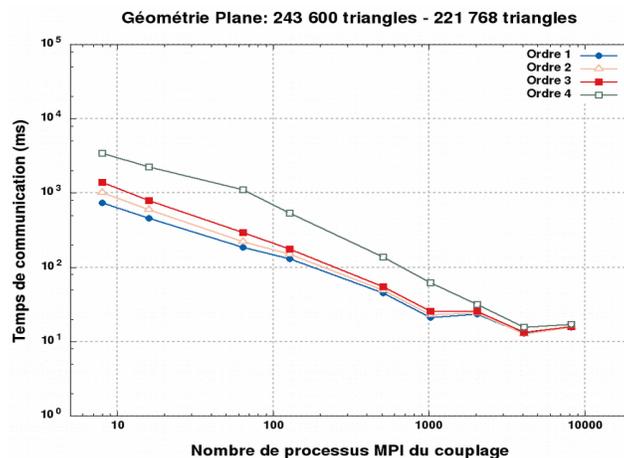


Figure 3 : Scalabilité sur maillage plan – Temps d'interpolation et d'échange d'un champ

## 2.2 Surface courbe

La surface d'échange sphérique est maillée de part et d'autre par 963 496 et 883 944 triangles. 60 millions de points sont localisés, interpolés et échangés à travers chacun des maillages.

Jusqu'à environ 500 cœurs de calcul la Figure 4 montre que la courbe de scalabilité du temps de localisation des maillages d'ordre 2 à 4 a une meilleure tendance que pour l'ordre 1. La tendance est ensuite la même quelque soit l'ordre. Le fort surcoût lié à l'ordre sur un petit nombre de cœurs n'est pas immédiatement explicable. Il faudrait faire des tests supplémentaires en faisant varier la tolérance géométrique de l'algorithme et la densité de points utilisé pour le calcul des boîtes englobantes des éléments pour en comprendre exactement l'origine.

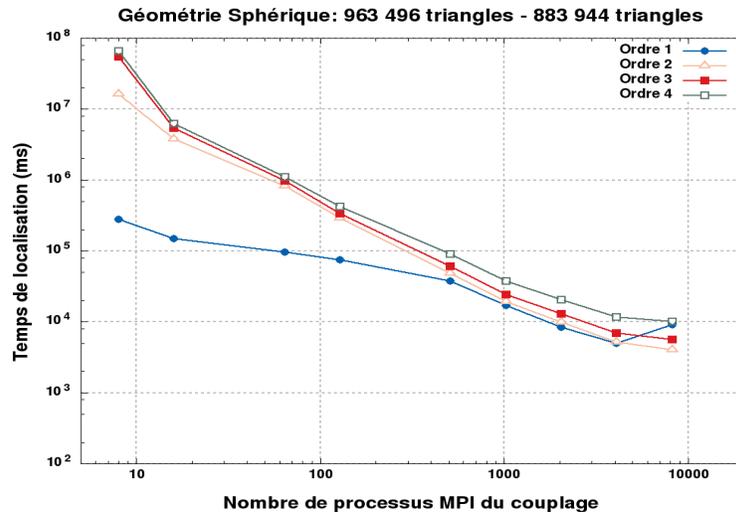


Figure 4 : Scalabilité sur maillage sphérique – Temps de localisation

Concernant le temps d'interpolation et d'échange du champ, la Figure 5 montre tout comme pour le cas plan que quelque soit l'ordre la courbe de scalabilité conserve la même tendance. Le nombre de points à traiter étant quatre fois plus important que pour le cas plan, le temps de calcul d'interpolation reste prédominant par rapport au temps d'échange quelque soit le nombre de cœurs de calcul utilisé.

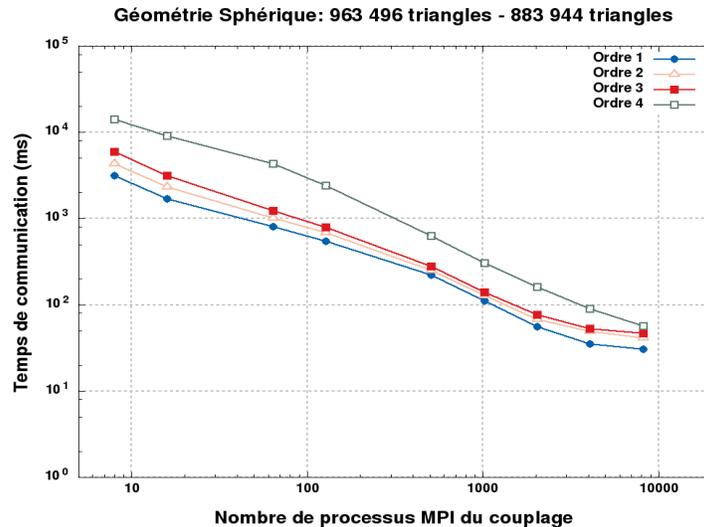


Figure 5 : Scalabilité sur maillage sphérique – Temps d’interpolation et d’échange d’un champ

### 3. INTÉGRATION DANS OPENPALM

L’utilisation de CWIPI dans OpenPALM passe par une couche d’interface des primitives CWIPI, l’enrichissement de l’API de CWIPI pour traiter les éléments d’ordre élevé a donc nécessité d’interfacer les nouvelles primitives dans OpenPALM. Ce travail est réalisé dans le fichier PWC.c dans le répertoire PrePALM\_MP/TEMPLATE de la distribution d’OpenPALM.

#### 3.1. Interface pour les codes C/C++

Quatre primitives concernant les éléments d’ordre élevé ont été ajoutées :

- PCW\_Ho\_define\_mesh                   => cwipi\_ho\_define\_mesh
- PCW\_Ho\_ordering\_from\_ijk\_set   => cwipi\_ho\_ordering\_from\_IJK\_set
- PCW\_Ho\_options\_set                 => cwipi\_ho\_options\_set
- PCW\_Ho\_set\_interpolation\_function => cwipi\_ho\_set\_interpolation\_function

Ces quatre primitives PCW\_ prennent strictement les mêmes arguments que les primitives cwipi\_. On peut cependant noter qu’avec OpenPALM le premier argument qui est le nom du couplage peut être différent entre les deux codes couplés, l’association entre ces deux noms de couplages se fait via l’interface graphique PrePALM, ce qui autorise une meilleure généricité pour l’instrumentation des codes. Ces primitives retournent un code d’erreur différent de 0 en cas d’échec.

#### 3.2. Interface pour les codes Fortran

L’API pour les codes Fortran (casse insensitive) pour OpenPALM repart de l’API C de CWIPI, les noms des fonctions pour l’API Fortran sont :

- pcw\_ho\_define\_mesh
- pcw\_ho\_ordering\_from\_ijk\_set
- pcw\_ho\_options\_set
- pcw\_ho\_set\_interpolation\_function

Les primitives étant des sous-routines Fortran (équivalent fonction void en C), un argument supplémentaire, par rapport à l'API C, en fin de liste permet de retourner le code d'erreur.

### 3.3. Interface pour les codes Python

L'API Python de CWIPI a elle aussi été complétée pour pouvoir traiter les éléments d'ordre élevé. Dans OpenPALM cette interface est faite le fichier cython `interface_pcw.pyx` du répertoire `PrePALM_MP/TEMPLATES`. Une fois l'objet de couplage défini l'utilisateur dispose des 4 méthodes suivantes pour activer les fonctions relatives aux éléments d'ordre élevé.

- `ho_define_mesh`
- `ho_ordering_from_ijk_set`
- `ho_options_set`
- `ho_set_interpolation_function`

Les tableaux à passer en argument sont de type NUMPY

### 3.4. Interface IP

OpenPALM permet de réaliser des couplages sur des machines distantes ou hétérogènes via un fonctionnement par serveur (miroir dans PALM) et client IP (code de calcul sur la machine distante). Pour C/C++ comme pour fortran, les primitives relatives aux éléments d'ordre élevé appelables coté client ont les mêmes arguments que pour une instrumentation directe. Seul le nom des primitives est différent :

- `PCWIP_Ho_define_mesh`                               => `cwipi_ho_define_mesh`
- `PCWIP_Ho_ordering_from_ijk_set`                   => `cwipi_ho_ordering_from_IJK_set`
- `PCWIP_Ho_options_set`                               => `cwipi_ho_options_set`

L'interface Python pour IP (`palmpip4py`) a également été construite.

### 3.5. Cas test de validation

Différents cas test de validation sont disponibles dans la release d'OpenPALM. Ces cas test sont en partie issus des cas test de CWIPI qui ont été portés sous OpenPALM. On trouvera ces tests dans les répertoires suivants :

- 
- `PALM_MP/TEST/c_cwipi_mpi1/cwipi_ho/c_surf_location_curvetriaP2/`       code C
- `PALM_MP/TEST/c_cwipi_mpi1/cwipi_ho/c_surf_location_triaP2`           code C
- `PALM_MP/TEST/c_cwipi_mpi1/cwipi_ho/py_surf_location_curvetriaP2/`   code python
- `PALM_MP/TEST/c_cwipi_mpi1/cwipi_ho/c_surf_gmsh`                       code C, gmsh
- `PALM_MP/TEST/z_ipilib/c_cwipi_mpi1/e_c_surf_coupling_simple_location/` interface IP
- `PALM_MP/TEST/z_ipilib/d_python/a_mpi_1/py_surf_location_curvetriaP2/` code python avec IP

Le cas test le plus représentatif se trouve dans le répertoire `c_surf_gmsh`.

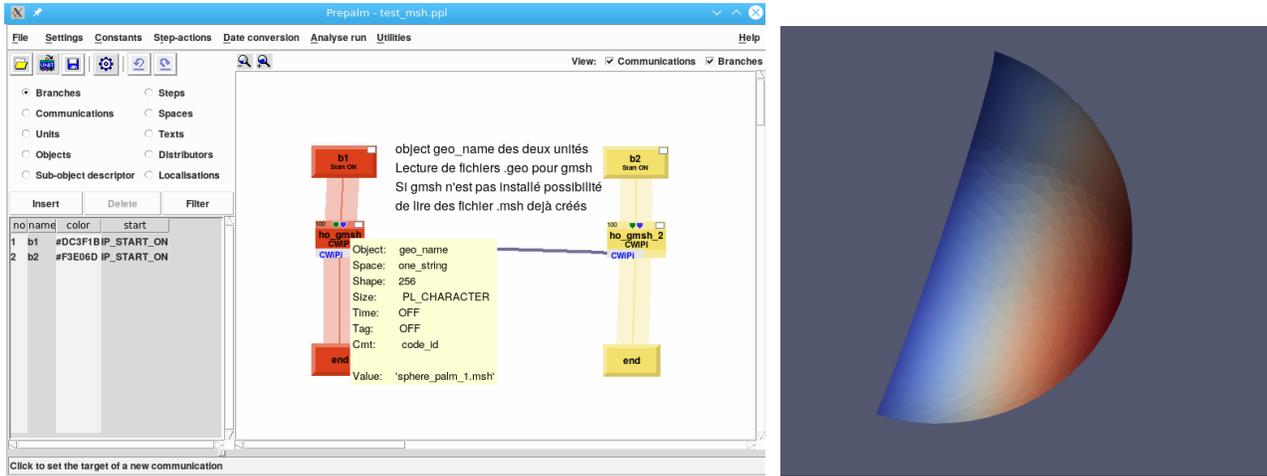


Figure 6 :IHM prepalm, couplage de deux ¼ de sphère à des résolution différentes

Ce cas test permet de coupler deux instances du programme test ho\_gmsh.c. Ce code permet de lire soit un fichier gmsh d'extension .geo, dans ce cas il construit le maillage en appelant le mailleur gmsh, soit de lire un fichier .msh déjà construit. Le choix des fichiers de maillage en entrée de chaque code (.geo ou .msh) se fait dans l'interface graphique prepalm en renseignant le plot d'entrée de l'objet geo\_name. Dans la distribution d'OpenPALM, deux maillages correspondant à ¼ d'une sphère à différentes résolutions sont fournis au format .geo et au format .msh. Le fichier prepalm test\_msh.ppl donne lit des fichiers .msh où le maillage est déjà construit.

### 3.6. Compatibilité des versions

Les éléments d'ordre élevé sont disponibles à partir de la version OpenPALM 4.3.0, dont la distribution est téléchargeable sur le site WEB d'OpenPALM à cette adresse : [https://www.cerfacs.fr/globc/PALM\\_WEB/](https://www.cerfacs.fr/globc/PALM_WEB/). Cette version d'OpenPALM comprend la distribution 0.10.3 de CWIPI ; les anciennes versions de CWIPI ne sont plus compatibles avec cette version d'OpenPALM.

## 4. APPLICATION AU COUPLAGE CEDRE/SPACE AEROACOUSTIQUE

Bien que développé en dehors du CERFACS cette exemple d'utilisation concrète, nous a semblé intéressant à montrer pour se rendre compte des possibilités applicatives des nouvelles fonctionnalités de CWIPI.

Le bruit de jet est un sujet d'étude intéressant l'ONERA [2][3][4]. Ces études peuvent notamment être réalisées par couplage de deux codes. Le premier est un code de CFD qui simule la génération des sources acoustiques et le second assure la propagation du champ acoustique dans les parties les plus lointaines du domaine d'étude.

Le cas d'étude retenu concerne l'environnement acoustique autour du pas de tir d'un lanceur lors de son décollage [3][4]. Le code CFD utilisé est CEDRE, il simule la combustion et permet de générer les sources acoustiques associées. Il se base des méthodes numériques de type volumes finis d'ordre 2. Le code de propagation acoustique est SPACE, il se base sur des méthodes numérique de type Galerkin

discontinu avec adaptation locale de l'ordre. SPACE a la capacité de traiter les maillages d'ordre élevé géométrique courbe. La nouvelle fonctionnalité de CWIPI permet donc de fournir à SPACE un maillage de ce type. Ce maillage (Figure 8) a été fourni par l'équipe Gamma 3 de l'INRIA en transformant le maillage initial (Figure 7).

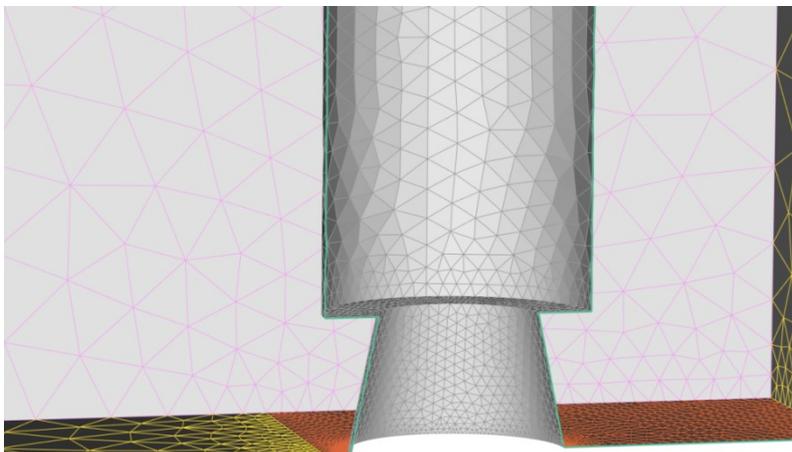


Figure 7 : Application aéroacoustique – Maillage initial SPACE

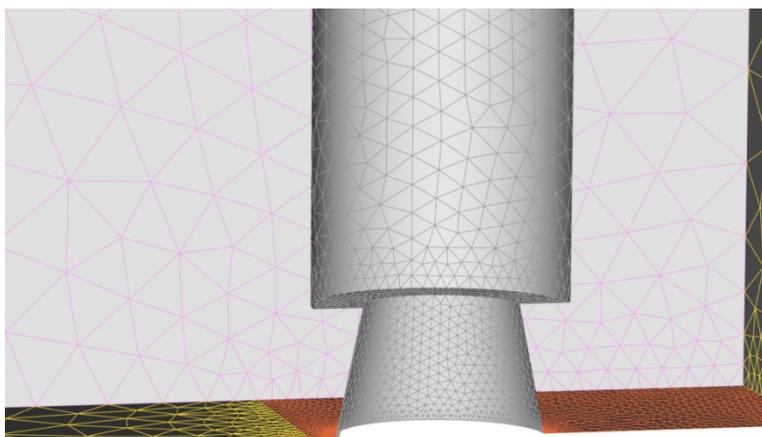


Figure 8 : Application aéroacoustique – Nouveau maillage SPACE d'ordre 2

La Figure 9 montre que la fonctionnalité ordre élevé peut d'ore et déjà être utilisée sur des configurations industrielles. Le champ d'application devra être étendu par la définition d'interfaces de couplage plus complexes

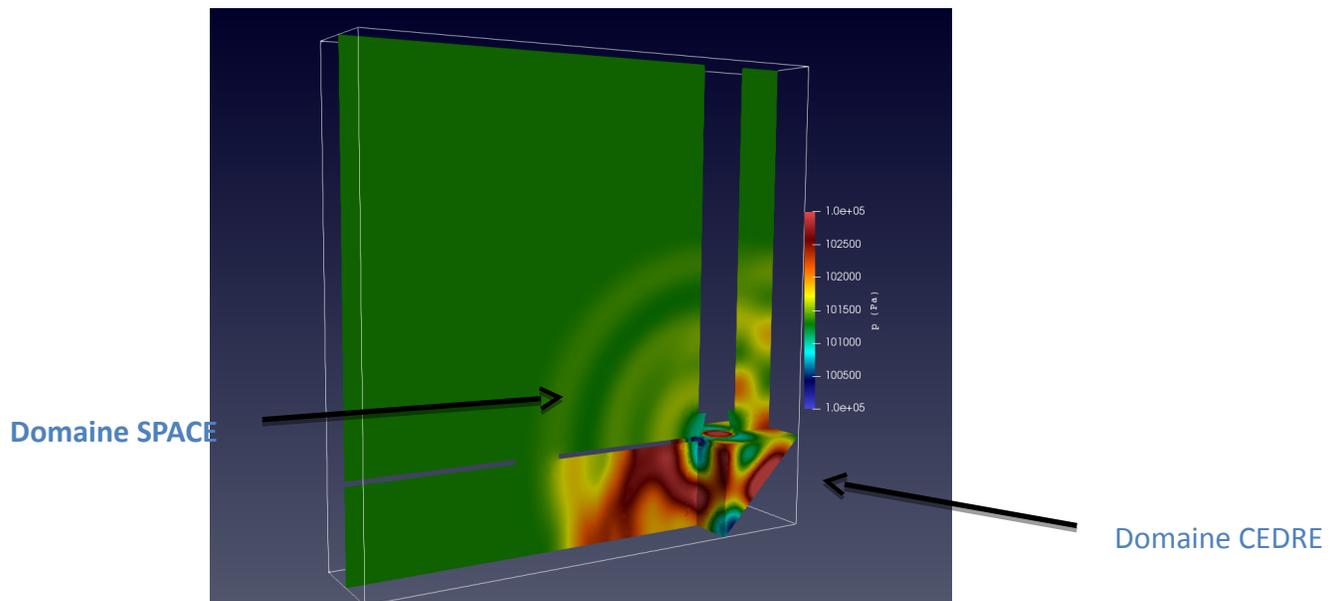


Figure 9 : Application aéroacoustique – Résultats

## 5. PERSPECTIVES

Voici les prochaines évolutions envisagées pour la fonctionnalité couplage ordre élevé de CWIPI :

- Prendre en compte les éléments linéiques et volumiques d'ordre élevé.
- Intégrer cette nouvelle fonctionnalité dans la nouvelle branche majeure de CWIPI développée au CERFACS. L'algorithme bénéficiera ainsi de l'équilibrage dynamique des nouveaux arbres de recherche et des opérations élémentaires géométriques. Ceci le rendra nettement plus efficace.
- Permettre à l'utilisateur de définir des éléments dont les nœuds ne sont pas répartis uniformément sur l'élément. La localisation des points d'interpolation est optimisée afin de limiter le développement du phénomène de Runge avec la montée en ordre (augmentation de l'amplitude des oscillations des polynômes d'interpolation sur les bords).
- Proposer une visualisation des interfaces d'ordre élevé. Le format Ensight Gold actuellement utilisé par CWIPI ne couvre pas le besoin. Il faut se tourner vers un autre format.
- Poursuivre les travaux sur les cas industriels d'ordre élevé

## 6. RÉFÉRENCES

- [1] R. Léger, C. Peyret, and S. Piperno. Coupled discontinuous galerkin/finite difference solver on hybrid meshes for computational aeroacoustics. *AIAA Journal*, 50:338–349, February 2012.
- [2] O. Labbé, C. Peyret, G. Rahier, and M. Huet. A CFD/CAA coupling method applied to jet noise prediction. *Computers and Fluids*, 86(C):1–13, Nov. 2013.

- [3] A. Langenais, F. Vuillot, C. Peyret, G. Chaineray, and C. Bailly. Assessment of a two-way coupling methodology between a flow and a high-order nonlinear acoustic unstructured solvers. *Flow, Turbulence and Combustion*, Jun 2018.
- [4] A. Langenais, F. Vuillot, J. Troyes, and C. Bailly. Accurate simulation of the noise generated by a hot supersonic jet including turbulence tripping and nonlinear acoustic propagation. *Physics of Fluids*, 31(1):016105, 2019.
- [5] F. Duchaine, S. Jauré, D. Poitou, E. Quémerais, Analysis of high performance conjugate heat transfer with the OpenPALM coupler, *Comput. Sci. Discovery* 8, 2015
- [6] Lackhove, K., Sadiki, A., & Janicka, J. (2017). Efficient Three Dimensional Time-Domain Combustion Noise Simulation of a Premixed Flame Using Acoustic Perturbation Equations and Incompressible LES. In *Proceedings of ASME Turbo Expo 2017*.
- [7] An Efficient LES-Acoustic Coupling Method for Sound Generation and High Order Propagation from Jets M.A. Moratilla-Vega, Kilian Lackhove, Johannes Janicka, H. Xia and G.J. Page, *ICCFD10*, 2018
- [8] C. Geuzaine and J.-F. Remacle. *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*. *International Journal for Numerical Methods in Engineering* 79(11), pp. 1309-1331, 2009.