CSMA 2024
16ème Colloque National en Calcul des Structures
13-17 Mai 2024, Presqu'île de Giens (Var)

pyVTCR : une nouvelle librairie pour la simulation du couplage vibro-acoustique en moyenne fréquence.

Thévenot[1], Ta[1], Puel[1], Barbarulo[1]

[1] *Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay, 91190, Gif-sur-Yvette, France*

Résumé — pyVTCR est une nouvelle librairie python à vocation open source ayant pour objectif la simulation de problèmes vibro-acoustique en moyenne fréquence. Cette librairie utilise la théorie variationelle des rayons complexes afin de s'affranchir de l'erreur de dispersion et des coûts de calcul grandissants inhérents à la méthode des éléments finis classique. De plus, l'architecture du code a été conçue afin de permettre l'utilisation de librairies python pour le calcul haute performance en étant hautement parallélisable et pouvant même utiliser le calcul GPU pour la résolution des systèmes linéaires. pyVTCR est en cours de développement dans le cadre d'une collaboration entre le LMPS et la SNCF. Après une présentation générale du contexte du développement du code, nous présentons l'architecture générale du code pyVTCR pour finir sur quelques exemples d'utilisation.
Mots clés — TVRC, moyenne fréquence, couplage vibro-acoustique, pyVTCR.

1 Contexte du développement de pyVTCR

Une des principales difficultés de la simulation vibro-acoustique en moyenne fréquence est l'explosion du coût de calcul lorsque la fréquence augmente. Deraemaeker [4] montre l'existence d'une erreur de pollution en plus de l'erreur de discrétisation de la méthode des éléments finis pour la résolution de l'équation de Helmholtz. Cette erreur limite grandement l'utilisation de la méthode pour les hautes et moyennes fréquences, c'est pourquoi de nombreuses techniques ont été développées au cours des dernières années comme résumé dans le livre *Methodologies for Mid-Frequency Analysis in Vibration and Acoustic* [1]. C'est dans ce contexte que P. Ladevèze [5] introduit en 1996 la théorie variationnelle des rayons complexes (TVRC), une méthode dite de Trefftz. Le code pyVTCR est l'implémentation la plus récente de la TVRC, son développement a débuté en 2023 par Raphael Thevenot et Antoine Solcourt, à la suite des travaux de Nhat Quang Ta dans le cadre de sa thèse entre le LMPS et la SNCF. L'objectif de pyVTCR est multiple :

- Créer une platforme unifiant les 20 ans de développement de la TVRC (en particulier pour la simulation acoustique et vibratoire).
- Tirer parti du langage python afin de profiter du dynamisme de sa communauté et des récentes librairies de calcul haute performance.
- Utiliser la programmation orientée objet afin de créer une architecture modulable et suffisament souple pour être utilisée dans des travaux de thèse, pour des projets pédagogiques et de nouvelles collaborations industrielles.
- Permettre l'ajout de nouvelles fonctionnalités dans le futur et la pérennité du développement.

Aujourd'hui le code prend en charge les problèmes d'acoustiques 2D [3] avec la possibilité de faire du couplage vibro-acoustique avec des paroies fines. Des travaux actuels ont pour but d'ajouter la résolution de problèmes de vibrations de coques orthotropes [2], d'autres s'intéressent à l'estimation d'erreur (thèse de Nhat Quang Ta en cours). Une prochaine thèse en partenariat entre le LMPS et la SNCF a vocation à développer la TVRC pour le couplage vibro-acoustique entre un milieu élastique et une cavité acoustique et d'implémenter une méthode hybride entre la TVRC et les éléments finis.

2 Présentation du code pyVTCR

Le code pyVTCR se fonde sur la TVRC pour résoudre des problèmes de vibrations et d'acoustique. Dans un premier temps, la TVRC est présentée dans le cadre de l'acoustique 2D puis l'architecture globale du code est décrite.

2.1 La TVRC appliquée à l'acoustique 2D

La figure 1 présente un cas de référence de problème acoustique ou le domaine Ω étudié est scindé en 2 sous-domaines. Dans la suite, nous considérerons un domaine partitionné en E sous-domaines différents.

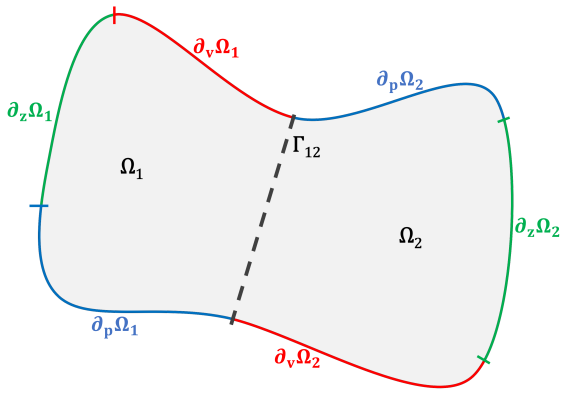


FIGURE 1 – Illustration d'un problème typique d'acoustique 2D.

Une première étape nécessaire à la TVRC est la partition du domaine étudié en E sous-domaines dits étoilés [6] :
Soit Ω le domaine étudié et $\{\Omega_e\}_{1\leq e\leq E}$ une partition de Ω telle que chaque sous-domaine Ω_e est étoilé.

$$\Omega = \bigcup_{1\leq e\leq E} \Omega_e, \Omega_e \cap \Omega_f = \emptyset \text{ if } e \neq f$$

Un domaine Ω_e est dit étoilé si et seulement si :

$$\exists\, a \in \Omega_e \mid \forall x \in \Omega_e, \{(1-t)a + tx \mid t \in [0,1]\} \subset \Omega_e$$

La méthode TVRC étant une méthode de résolution fréquentielle, on fixe une fréquence d'étude f et le champs de pression acoustique p_e, défini sur Ω_e, vérifie l'équation d'Helmholtz suivante :

$$\Delta p_e + k^2 p_e = 0 \quad \text{dans } \Omega_e \tag{1}$$

Avec $k = (1+i\eta)\frac{2\pi f}{c_o}$ le nombre d'onde supposé homogène, η, c_0 sont le facteur d'amortissement et la vitesse du son dans le milieu étudié.
Chaque sous-domaine peut être soumis à 3 types de conditions limites différentes :

- Condition de Dirichlet : $p_e = p_{imp}$ sur $\partial_p\Omega_e$
- Condition de Neumann : $\frac{i}{\rho_0 2\pi f}\frac{\partial p_e}{\partial \underline{n}_e} = v_{imp}$ sur $\partial_v\Omega_e$
- Condition de Robin : $p_e - \frac{iZ_{imp}}{\rho_0 2\pi f}\frac{\partial p_e}{\partial \underline{n}_e} = 0$ sur $\partial_Z\Omega_e$

De plus, la décomposition en sous-domaines introduit des conditions limites de continuité à la frontière Γ_{ef} entre les 2 sous-domaines Ω_e et Ω_f :

$$p_e - p_f = 0 \qquad \text{sur } \Gamma_{ef} = \partial\Omega_e \cap \partial\Omega_f \tag{2}$$

$$\frac{i}{\rho_0 2\pi f}\frac{\partial p_e}{\partial \underline{n}_e} + \frac{i}{\rho_0 2\pi f}\frac{\partial p_f}{\partial \underline{n}_f} = 0 \qquad \text{sur } \Gamma_{ef} = \partial\Omega_e \cap \partial\Omega_f \tag{3}$$

Les normales n_e et n_f sont sortantes de leur sous-domaine respectif.

Une fois le problème de référence correctement défini, on peut montrer que pour chaque sous-domaine étoilé Ω, la solution au problème peut s'écrire sous la forme d'une fonction d'Herglotz [6] :

$$p_e(\underline{x}) = \int_0^{2\pi} A_e(\theta) e^{i\underline{k}_e(\theta)\underline{x}} d\theta \tag{4}$$

Il s'agit d'une somme continue d'ondes planes de vecteur d'onde $\underline{k}_e(\theta)$ tel que $||\underline{k}_e|| = k$. Afin de résoudre numériquement ce problème, la méthode TVRC propose d'approximer cette somme continue par une somme discrète et finie. Plusieurs choix ont été étudié, mais le choix retenu est d'approximer la fonction d'amplitude A_e par une somme de dirac, on obtient donc l'approximation suivante :

$$p_e(\underline{x}) \approx \sum_{n=1}^{N} a_n e^{i\underline{k}_e(\theta_n)\underline{x}} \tag{5}$$

Cette formulation à l'avantage de faciliter l'implémentation numérique et d'accélérer les temps de calcul en permettant un calcul analytique des intégrales de la forme faible 6.
De par ce choix, la pression p_e vérifie *a priori* l'équation d'équilibre 1, il suffit donc de trouver les amplitudes a_n qui satisfont au mieux les conditions limites. Pour ce faire, on introduit la formulation faible du problème aux conditions limites :

Trouver $(p_1, ..., p_E) \in S_{1,ad} \times ... \times S_{E,ad}$, tel que :

$$\sum_{e=1}^{E} \Re\left(\int_{\partial_p\Omega_e} (p_e - p_{imp})\overline{\delta v_e} dS + \int_{\partial_v\Omega_e} (v_e - v_{imp})\overline{\delta p_e} dS + \int_{\partial_Z\Omega_e} (p_e - Z_{imp} v_e)\overline{\delta v_e} dS \right.$$
$$\left. + \sum_{f=1, f\neq e}^{E} \int_{\Gamma_{ef}} (p_e - p_f)\overline{\delta v_e} + (v_e + v_f)\overline{\delta p_e} dS \right) = 0 \tag{6}$$
$$\forall (\delta p_1, ..., \delta p_E) \in S_{1,ad} \times ... \times S_{E,ad}$$

Avec $v_e = \frac{i}{\rho_0 \omega} \frac{\partial p_e}{\partial \underline{n}_e}$, $\delta v_e = \frac{i}{\rho_0 \omega} \frac{\partial \delta p_e}{\partial \underline{n}_e}$ et $\overline{(\square)}$ qui correspond au conjugué complexe de $\square$.

A partir de cette forme, on peut introduire les sous-opérateurs suivants :

$$\mathcal{B}_p^e(p_e, \delta p_e) = \int_{\partial_p\Omega_e} p_e \overline{\delta v_e} dS \quad ; \quad \mathcal{B}_v^e(p_e, \delta p_e) = \int_{\partial_v\Omega_e} v_e \overline{\delta p_e} dS \quad ; \quad \mathcal{B}_Z^e(p, \delta p_e) = \int_{\partial_Z\Omega_e} (p_e - Z_{imp} v_e)\overline{\delta v_e} dS$$

$$\mathcal{B}_{\Gamma_f}^e(p_e, \delta p_e) = \int_{\Gamma_{ef}} p_e \overline{\delta v_e} + v_e \overline{\delta p_e} dS \quad ; \quad \mathcal{B}_f^e(p_f, \delta p_e) = \int_{\Gamma_{ef}} -p_f \overline{\delta v_e} + v_f \overline{\delta p_e} dS$$

$$\mathcal{L}_p(\delta p_e) = \int_{\partial_p\Omega_e} p_{imp} \overline{\delta v_e} dS \quad ; \quad \mathcal{L}_v(\delta p_e) = \int_{\partial_v\Omega_e} v_{imp} \overline{\delta p_e} dS$$

Ce qui permet de réécrire l'équation 6 sous la forme suivante :

$$\sum_{e=1}^{E} \Re\left(\mathcal{B}_p^e(p_e, \delta p_e) + \mathcal{B}_v^e(p_e, \delta p_e) + \mathcal{B}_Z^e(p_e, \delta p_e) + \mathcal{L}_p(\delta p_e) + \mathcal{L}_v(\delta p_e) + \sum_{f=1, f\neq e}^{E} (\mathcal{B}_{\Gamma_f}^e(p_e, \delta p_e) + \mathcal{B}_f^e(p_f, \delta p_e)) \right) = 0 \tag{7}$$

Ce qui donne le système linéaire suivant après discrétisation :

$$\underbrace{\begin{bmatrix} \mathbf{B}_p^1 + \mathbf{B}_v^1 + \mathbf{B}_Z^1 + \mathbf{B}_{\Gamma_f}^1 & \cdots & \mathbf{B}_1^E \\ \vdots & \ddots & \vdots \\ \mathbf{B}_E^1 & \cdots & \mathbf{B}_p^E + \mathbf{B}_v^E + \mathbf{B}_Z^E + \mathbf{B}_{\Gamma_f^E} \end{bmatrix}}_{\mathbb{B}} \underbrace{\begin{bmatrix} \mathbf{a_1} \\ \vdots \\ \mathbf{a_E} \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} \mathbf{L}_p^1 + \mathbf{L}_v^1 \\ \vdots \\ \mathbf{L}_p^E + \mathbf{L}_v^E \end{bmatrix}}_{\mathbb{L}} \tag{8}$$

2.2 Architecture de la librairie

L'utilisation de la programmation orientée objets nous a permis de refléter au mieux dans l'implémentation la structure des objets mathématiques/physiques de la TVRC. Le code pyVTCR est séparé en plusieurs parties, chacune permettant de simuler un certiain type de physique : une première partie permet de gérerla géométrie du modèle étudié, une seconde permet d'implémenter totue la résolution d'un problème acoustique ect. La figure 2 présente le diagramme de classe de pyVTCR.

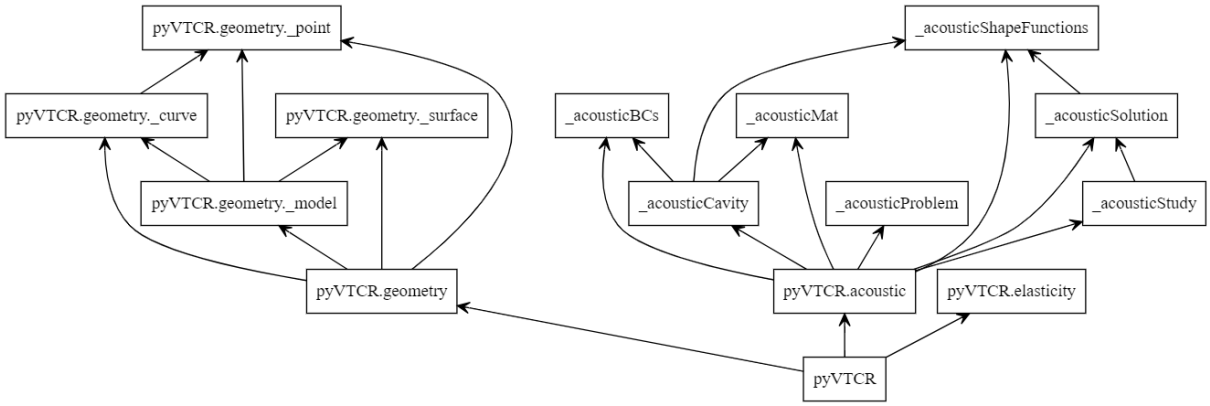
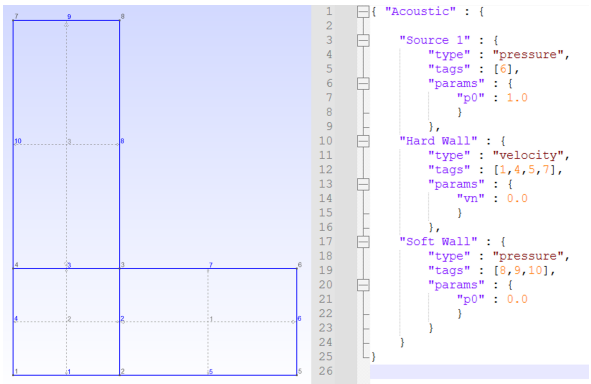


FIGURE 2 – Diagramme de l'organisation du code *pyVTCR*

La première étape pour réaliser l'étude d'un cas est de définir la géométrie du problème et ses conditions limites. Pour cela, pyVTCR utilise le module python Gmsh [7] qui permet la création et la manipulation de géométries complexes, en parallèle on définit les conditions dans un fichier .json qui sera lu par le code pyVTCR. La figure 3 présente un exemple de définition de problème acoustique.



```
{ "Acoustic" : {
    "Source 1" : {
        "type" : "pressure",
        "tags" : [6],
        "params" : {
            "p0" : 1.0
            }
        },
    "Hard Wall" : {
        "type" : "velocity",
        "tags" : [1,4,5,7],
        "params" : {
            "vn" : 0.0
            }
        },
    "Soft Wall" : {
        "type" : "pressure",
        "tags" : [8,9,10],
        "params" : {
            "p0" : 0.0
            }
        }
    }
}
```

FIGURE 3 – Exemple de définition d'un problème d'acoustique avec pyVTCR.

Ensuite la résolution du problème s'effectue en 4 étapes :

1. Création des fonctions de forme (ondes planes) en définissant une fréquence et leur distribution.
2. Une boucle sur les différents sous-domaines et leurs différentes conditions limites permet de créer les sous-opérateurs associés.
3. Après assemblage de l'opérateur global, on résoud le système en utilisant une pseudo inverse à cause du mauvais conditionnements de la matrice.
4. On peut projeter le résultat sur un maillage pour visualisation ou directement utiliser les fonctions de forme de la TVRC pour le post-traitement.

2.3 Quelques propriétés de la TVRC

La première propriété de la TVRC est qu'elle ne nécessite **pas de maillage**. Elle demande beaucoup moins de degrés de liberté pour résoudre un problème. De plus, l'utilisation des ondes planes comme fonctions de forme permet d'obtenir une solution C^∞ et ne nécessite aucune interpolation. Ces ondes

planes permettent un calcul analytique des intégrales nécessaire à la création du système à résoudre, ce qui permet de réduire significativement les temps de calcul.
Comme de nombreuses autres méthodes de Trefftz, le système à résoudre est mal conditionné, on peut noter les propriétés suivantes :

- La matrice $\mathbb{B}$ dépend fondamentalement de la fréquence étudiée et elle est mal conditionnée.
- La matrice $\mathbb{B}$ est dense par bloc (Cf. figure 4), elle est non symétrique, et à valeurs complexes.
- Les sous-matrices de couplage $\mathbf{B}_e^f$ sont nulles si les domaines e et f sont séparés, sinon elles sont pleines.

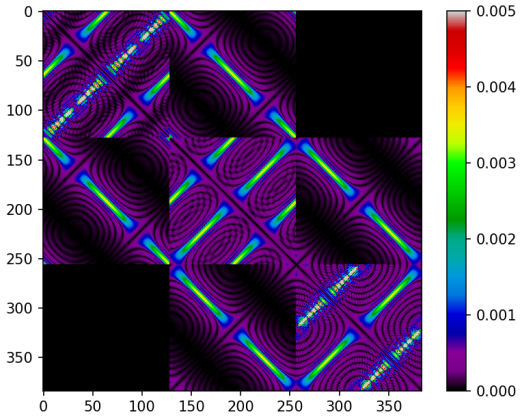


FIGURE 4 – Exemple d'une matrice TVRC pour l'exemple présenté figure 1.

3 Exemple d'utilisation et comparaison aux éléments finis

3.1 Exemple simple avec solution analytique

On propose d'étudier le cas d'une cavité acoustique rectangulaire avec pression imposée non nulle à droite et pression imposée nulle sur les autres bords. De plus, cette cavité est séparée en 2 sous-domaines distincts afin d'illustrer le couplage entre cavités. La figure 7 présente la géométrie complète et rappelle la solution analytique associée. Les résultats de l'utilisation du code pyVTCR pour cette simulation sont présentés figure 6.

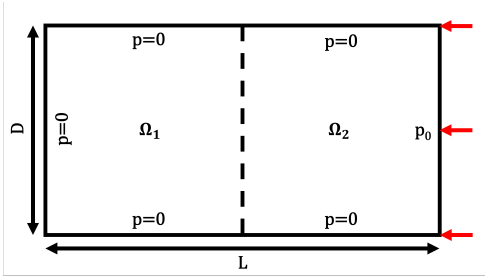


FIGURE 5 – Schéma d'une cavité acoustique soumise à un chargement en pression à droite.

Solution analytique :

$$p(x,y) = 4p_0 \sum_{n=0}^{\infty} \frac{\sin(k_{y,2n+1}y)\sin(k_{x,2n+1}x)}{(2n+1)\pi \sin(k_{y,2n+1}L)}$$

$$k_{y,n} = \frac{n\pi}{D} \quad ; \quad k_{x,n} = \sqrt{k^2 - k_{y,n}^2}$$

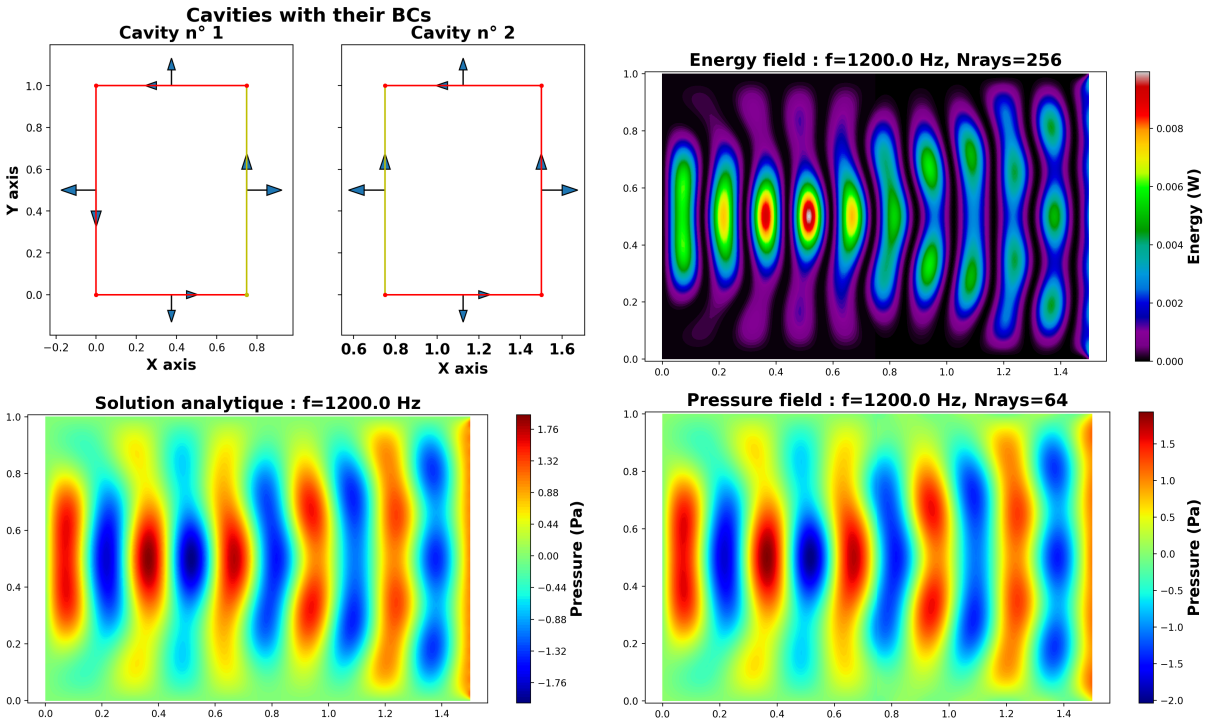


FIGURE 6 – Résultat de la méthode TVRC sur l'exemple présenté figure 7. En haut à gauche, géométrie des deux cavités avec leurs conditions limites (pression imposée en rouge, continuité en jaune). En haut à droite, énergie acoustique à l'intérieur des deux cavités. En bas, comparaison des champs de pression analytique et TVRC.

Nous avons ensuite pu comparer les résultats obtenue en utilisant pyVTCR avec ceux obtenus en utilisant le logiciel commercial Comsol. Les paramètres de la simuations sont présentés ci-dessous :

- Domaine de l'étude : entre 100 et 5000 Hz avec un pas de 10 Hz.
- Méthode éléments finis avec Comsol : 120k degrés de liberté (étude de convergence à 1% d'erreur, 15 éléments par longueur d'onde), temps de calcul : **422 s**.
- pyVTCR : 128 degrés de liberté, temps de calcul : **1 s**.

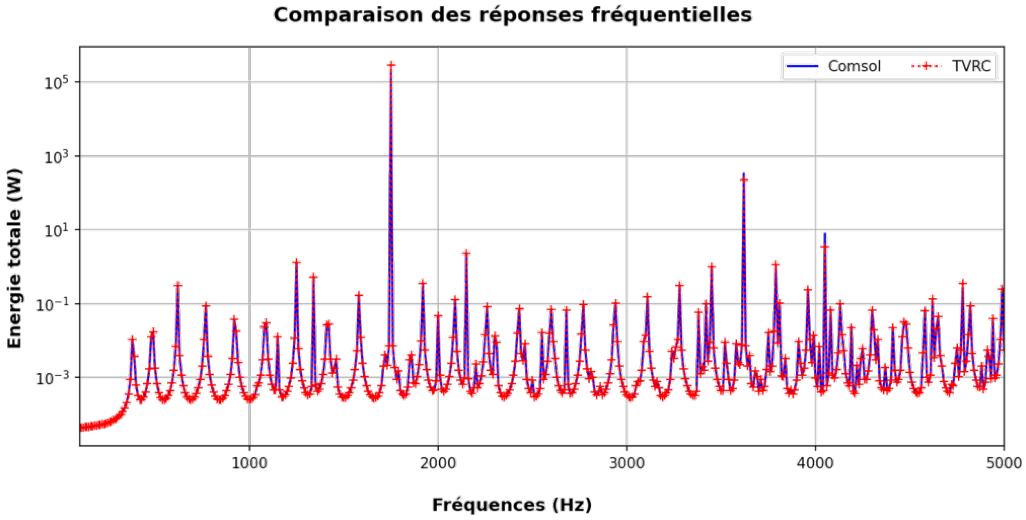
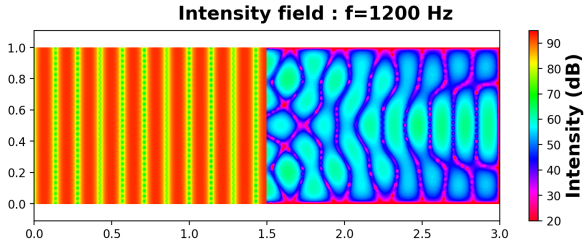


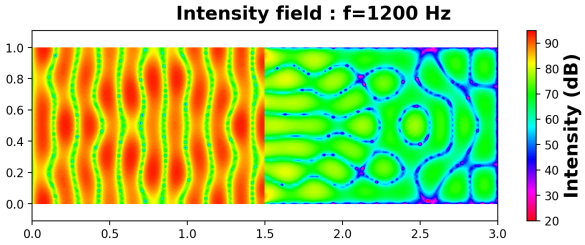
FIGURE 7 – Comparaison des réponses fréquentielles des deux méthodes.

3.2 Exemple de couplage entre 2 cavités acoustiques au travers d'une paroie fine

pyVTCR permet entre autre chose de simuler le couplage entre une cavité acoustique et une paroie fine. La figure 8 compare les résultats de deux simulations modélisant la transmission d'une onde acoustique à travers une paroie fine, la source acoustique est située sur la gauche des figures.



(a) Intensité acoustique dans les cavités séparées par une paroi en verre ($\rho = 2320 kg/m^3$, $E = 75\,GPa$, $\nu = 0.25$).

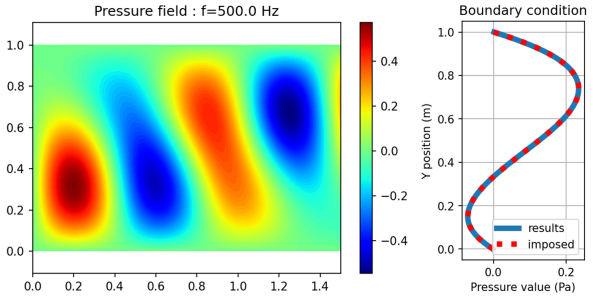


(b) Intensité acoustique dans les cavités séparées par une paroi en plexiglas ($\rho = 1180 kg/m^3$, $E = 3\,GPa$, $\nu = 0.4$).

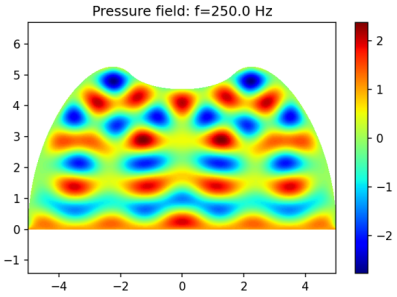
FIGURE 8 – Comparaison du couplage entre deux cavités séparées par une paroi.

3.3 Aperçu des développements futures de pyVTCR

Comme expliqué précédemment, le code pyVTCR est en cours de développement et de nombreuses nouvelles fonctionnalités sont à venir. Les figures suivantes présentent les récentes améliorations du logiciel.



(a) Integration semi-analytique de conditions limites polynomiales. Ici chargement en pression sur le bord droit.



(b) Possibilité de définir des bords quelconques.

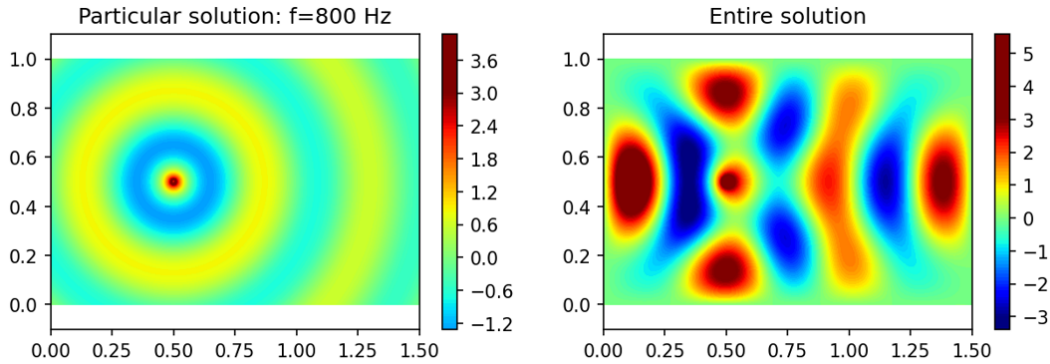


FIGURE 10 – Utilisation de sources ponctuelles à l'intérieur du domaine.

Références

[1] O. Atak, B. Pluymers, W. Desmet et al., *CAE Methodologies for Mid-Frequency Analysis in Vibration and Acoustics*, 2012.

[2] A. Cattabiani, A. Barbarulo, et al, *Variational theory of complex rays applied to shell structures : in-plane inertia, quasi-symmetric ray distribution, and orthotropic materials*, Computational Mechanics, 983-997, 2015.

[3] R. Cettour-Janet, *Modelling the vibrational response and acoustic radiation of the railway tracks*, Thèse de doctorat, Université Paris Saclays, 2019.

[4] A. Deraemaeker, I. Babuska, P. Bouillard. *Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions*, Int. J. Numer. Meth. Engng., 471-499, 1999.

[5] P. Ladevèze, *Une nouvelle approche pour le calcul des vibrations moyennes fréquences*, compte rendu de l'académie des sciences de Paris, 849-856, 1996.

[6] R. L. Ochs, *A version of Runge's theorem for the Helmholtz equation with applications to scattering theory*, Proceedings of the Edinburgh Mathematical Society, 107-119, 1989

[7] 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.