

EI Contrôle de la pollution acoustique intérieure

Thomas Lemerrier, Timothé Boulet, Elena López, Yanis Adel, Paul Castéras

15 Novembre 2021

1 Introduction

L'objectif de notre start-up Liners Company est de contrôler et de réduire le bruit émis par un réacteur d'avion. Une technique classique et très utilisée de nos jours est l'utilisation de liners, comme cela est visible sur la figure 1.

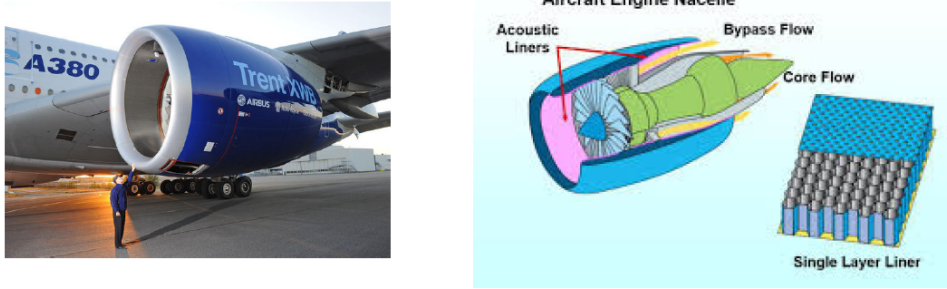


Figure 1: Utilisation de liners pour un avion.

Cependant, leur utilisation est relativement coûteuse et nous n'en disposons pas ici en quantité suffisante afin de pouvoir les répartir sur toute la surface disponible. Nous devons donc considérer une quantité fixée de liners, et déterminer leur répartition optimale afin de minimiser l'énergie acoustique au sein du réacteur. Le problème que nous allons traiter est alors celui d'un domaine avec une onde plane incidente, comme visible sur la figure 2.

Nous pouvons modéliser ce problème mathématiquement, et nous obtenons alors le problème d'Helmholtz visible sur la figure 3, qu'il s'agit de résoudre tout en essayant d'optimiser la solution obtenue. Ce problème étant complexe, nous ne résoudrons pas les équations analytiquement, mais nous effectuerons plutôt une résolution numérique des équations du problème dans la gamme de fréquences d'intérêt.

Les moteurs d'avion présentent des fréquences allant généralement de 1000 Hz à 4000 Hz, nous évaluerons les fréquences allant de 0 Hz à 4000 Hz.

L'objectif est alors de réduire le bruit de manière optimale, c'est-à-dire de déterminer où répartir les liners au sein du réacteur de manière optimale. Afin de traiter ce problème mathématiquement et numériquement, il est nécessaire d'également modéliser la répartition des liners. Nous la représentons alors par une fonction caractéristique χ définie sur la frontière Γ , telle que :

$$\chi(\mathbf{x}) = \begin{cases} 1 & \text{s'il y a un matériau poreux en } \mathbf{x} \\ 0 & \text{sinon} \end{cases} \quad (1)$$

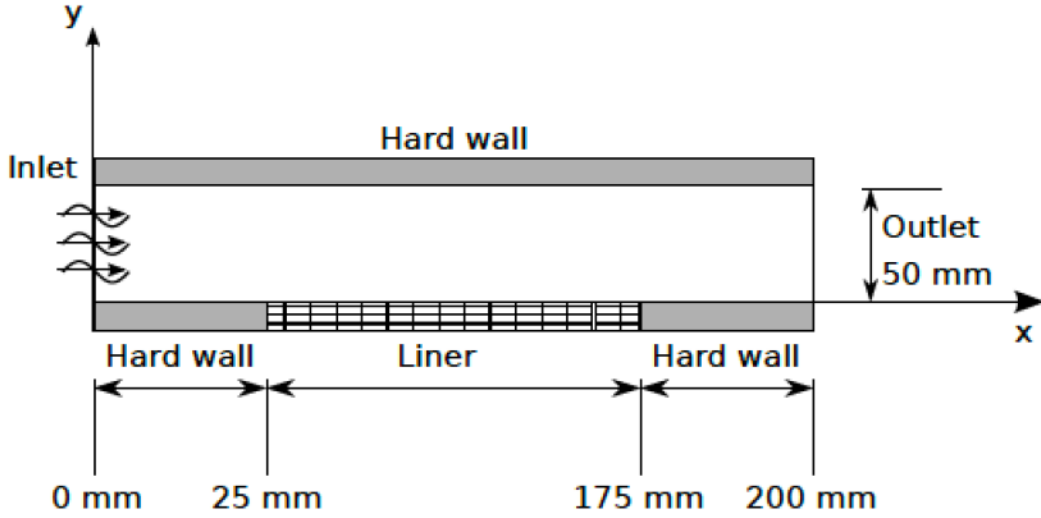


Figure 2: Schéma du problème à traiter.

avec la condition $0 < \int_{\Gamma} \chi dS < \beta$, qui représente la quantité maximale de liners dont nous disposons.

L'objectif du problème est alors de déterminer χ telle que la nuisance sonore notée $J(\chi)$, définie par :

$$J(\chi) = \int_{\Omega} |u_{\chi}|^2 d\mathbf{x}$$

est minimale, où u_{χ} est solution de :

$$\begin{cases} \Delta u + k^2 u = 0 & \text{dans } \Omega \\ u = g & \text{sur } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} = 0 & \text{sur } \Gamma_N \\ \frac{\partial u}{\partial \mathbf{n}} + \alpha \chi u = 0 & \text{sur } \Gamma \end{cases} \quad (2)$$

Afin de respecter la contrainte sur la quantité de liners, nous choisirons χ appartenant à l'ensemble :

$$U_{ad}(\beta) = \left\{ \chi \in L^{\infty}(\Gamma); \forall \mathbf{x} \in \Gamma, \chi(\mathbf{x}) \in \{0, 1\}, \int_{\Gamma} \chi dS = \beta \right\}$$

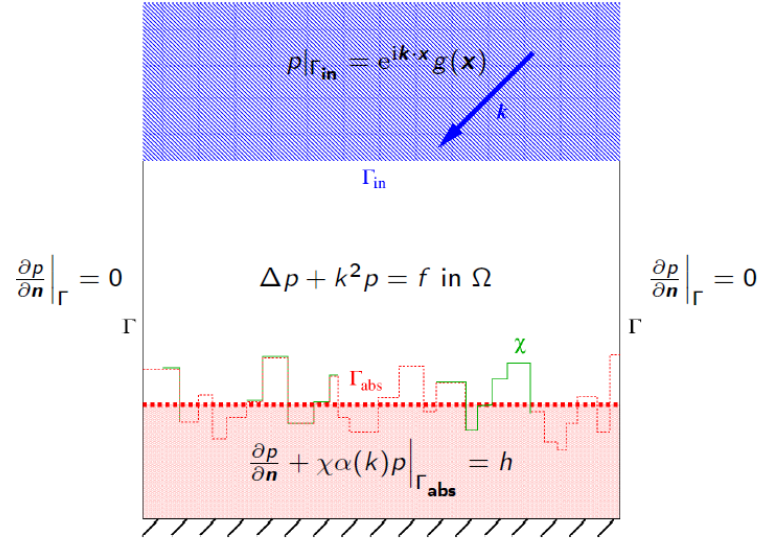
Mais en réalité, pour des soucis d'existence de minimum, on peut plutôt chercher χ dans l'ensemble :

$$U_{ad}(\beta) = \left\{ \chi \in L^{\infty}(\Gamma); \forall \mathbf{x} \in \Gamma, \chi(\mathbf{x}) \in [0, 1], \int_{\Gamma} \chi dS = \beta \right\}$$

2 Résolution numérique du problème

Nous effectuons ici une résolution numérique du problème par la méthode des différences finies. Ainsi, nous définissons un maillage de l'espace, qui peut de plus avoir une frontière irrégulière, dont nous pouvons choisir arbitrairement le niveau de l'irrégularité.

Pour des soucis pratiques, une fois le maillage fait, nous initialisons les valeurs des paramètres et des fonctions, et notamment de celles représentant les conditions aux limites en chaque point du maillage. La valeur du paramètre α est de plus calculée de manière



Model problem for acoustical liners with $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{abs}} \cup \Gamma$

- Γ_{in} inflow condition (e.g. incident plane wave with vanishing Mach number $M_0 = 0$);
- Γ hard-wall ($Z \rightarrow \infty$);
- Γ_{abs} acoustic liner with absorption coefficient $\alpha \in \mathbb{C}$.

Figure 3: Modélisation mathématique du problème à résoudre.

optimale pour une pulsation ω donné, pour le matériau considéré. On le définit ensuite en chaque point du maillage, où il vaut cette valeur optimale s'il y a un matériau poreux en ce point, et 0 sinon.

Les paramètres et les conditions aux limites étant alors définis en chaque point du maillage, nous résolvons alors le problème grâce à la méthode des différences finies. L'objectif est dès lors d'optimiser la solution obtenue u grâce à la fonction caractéristique χ afin de minimiser la valeur de J , ce qui fait l'objet de la section suivante.

3 Optimisation de la solution

3.1 Présentation du premier algorithme utilisé

Le problème consistant en la minimisation de la fonction $J(\chi)$, on effectue cette minimisation par la méthode des gradients projectés, avec comme fonction de coût $J(\chi)$ [FS21]. Le fonctionnement de l'algorithme est le suivant:

1. On considère au début $\chi^{(0)}$ choisi aléatoirement dans $\{0, 1\}^N$
2. On calcule lors de l'étape k , $\chi^{(k+1)} = P_{l_k}[\chi^{(k)} - \mu_k J'(\chi^{(k)})]$, où
 - $\mu_k > 0$ est la vitesse d'apprentissage, qui est mise à jour à chaque étape :
 - Si $J(\chi^{(k+1)}) < J(\chi^{(k)})$, alors $\mu_{k+1} = 1.1 * \mu_k$ pour la prochaine itération.
 - Si $J(\chi^{(k+1)}) \geq J(\chi^{(k)})$, alors $J(\chi^{(k+1)})$ est réévalué en divisant μ_k par 2.

- $P_{l_k}(\chi) = \max(0, \min(\chi + l, 1))$ est un projecteur, qui garantit que $\chi^{(k+1)} \in [0, 1]$, et où l est un facteur de Lagrange, qui assure que la contrainte $\int_{\Gamma} \chi dS = \beta$ est respectée. Il est mis à jour à chaque étape également.
- L'algorithme s'arrête après K itérations, ou dès que $\|\chi^{(k+1)} - \chi^{(k)}\|_2$ par exemple est suffisamment faible ce qui signifie que le gradient est faible et donc qu'on a atteint un minimum local.

L'algorithme en pseudo-code est visible sur la figure 4.

```

 $\chi^{(0)} \in U_{\text{ad}}(\beta); \mu = \mu^{(0)};$ 
for  $k = 0 : K$  do
  compute  $u^{(k)}; p^{(k)}; J(\chi^{(k)}); J'(\chi^{(k)});$ 
   $E = J(\chi^{(k)});$ 
  while  $E \geq J(\chi^{(k)})$  &  $\mu > \epsilon_0$  do
     $\ell = 0;$ 
     $\chi^{(k+1)} = \mathcal{P}_{\ell} [\chi^{(k)} - \mu J'(\chi^{(k)})];$ 
    while  $|\int_{\Gamma} \chi^{(k+1)} dS - \beta| \geq \epsilon_1$  do
      if  $\int_{\Gamma} \chi^{(k+1)} dS \geq \beta$  then
         $\ell \leftarrow \ell - \epsilon_2;$ 
      else
         $\ell \leftarrow \ell + \epsilon_2;$ 
      end if
       $\chi^{(k+1)} = \mathcal{P}_{\ell} [\chi^{(k)} - \mu J'(\chi^{(k)})];$ 
    end while
    compute  $u^{(k+1)}; J(\chi^{(k+1)});$ 
     $E = J(\chi^{(k+1)});$ 
    if  $E < J(\chi^{(k)})$  then
       $\mu \leftarrow \mu + \epsilon_3;$ 
    else
       $\mu \leftarrow \mu/2;$ 
    end if
  end while
end for

```

Figure 4: Pseudo-code de l'algorithme de descente de gradient.

3.2 Proposition d'algorithme

Une fois l'algorithme de la méthode des gradients projetés implémenté, nous avons implémenté d'autres algorithmes de minimisation pour pouvoir les comparer et sélectionner la meilleure méthode.

3.3 Deuxième algorithme

Le deuxième algorithme que nous proposons est identique à l'algorithme proposé en premier lieu, mais il réinitialise μ (Learning Rate) à sa valeur initiale après chaque descente de gradient.

Ce deuxième algorithme a alors une exécution au moins plus longue que le premier, mais permet théoriquement de plus "explorer" l'ensemble de solution. Comme le montre la figure 5.

Néanmoins en pratique, cette méthode n'améliorait pas le résultat final mais ajoutant du temps de calcul nous n'avons donc pas retenu cette méthode.

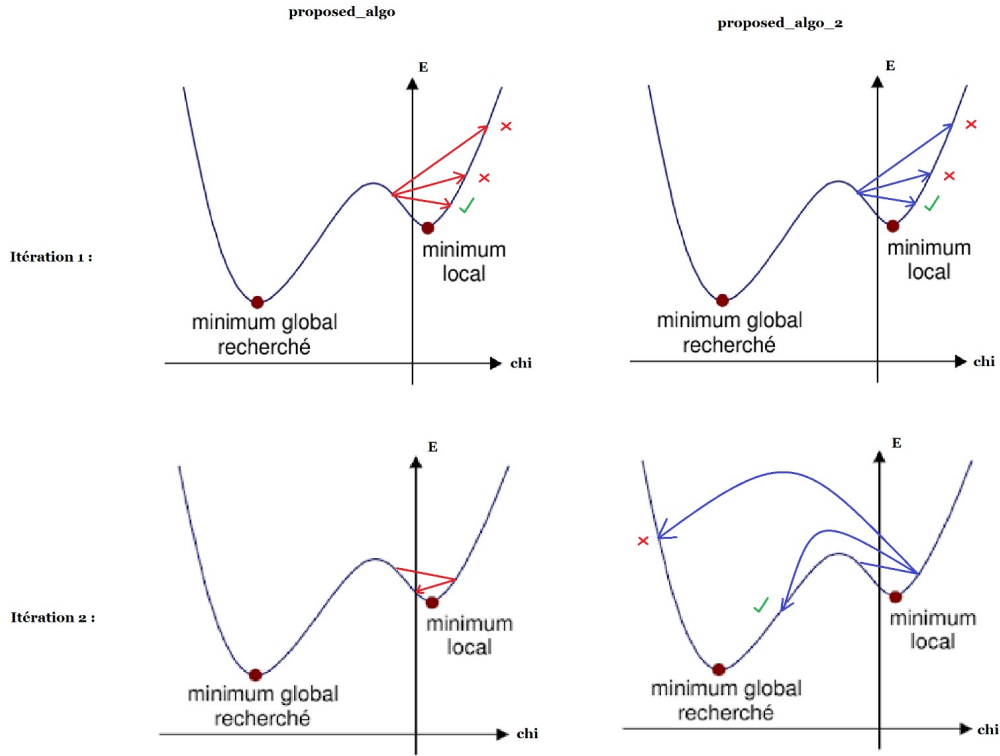


Figure 5: Comparaison entre les deux algorithmes.

3.4 Calcul du gradient par la limite

Une autre variante de l'algorithme consiste à calculer différemment le gradient de J . En effet dans le premier algorithme, nous avons implémenté une dérivée au sens de Frechet. Cependant, il existe d'autres façons de calculer des gradients et en particulier une définition classique de la dérivée est $\lim_{h \rightarrow 0} \frac{J(x+h) - J(x)}{h}$.

On peut donc définir le gradient de J en calculant les dérivées de J selon chaque composante de χ . Le gradient de l'énergie est alors un tableau de la taille du maillage qui est nul sauf sur la frontière. Et pour un élément du tableau de coordonnées (i,j) , on pose: $\nabla J[i,j] = \frac{J(P_l(\chi + h\delta_{i,j})) - J(P_l(\chi))}{h}$ avec $\delta_{i,j}$ un tableau nul en tout point sauf au point de coordonnées (i,j) égal à 1 et h un réel positif petit qui permet d'approximer la limite en zéro. L'intérêt est que l'on obtient un gradient de la vraie fonction à minimiser. L'inconvénient est évidemment que l'on doit recalculer l'énergie pour chaque composante ce qui se révèle très lourd en calcul.

3.5 Projection par softmax

En plus de modifier le comportement de l'algorithme d'un point de vue de la descente de gradient nous avons également essayer de modifier le projecteur P_l . Au lieu d'utiliser le projecteur $P_k(\chi) = \max(0, \min(\chi + l, 1))$ pour contraindre χ à être de somme β et de composantes entre 0 et 1, on peut utiliser une fonction softmax.

La fonction softmax est définie selon : $\sigma(z)_j = \frac{e^{z_j}}{\sum_{j=1}^N e^{z_j}}$ avec $z = [z_1, \dots, z_N]$ avec vecteur de \mathbb{R}^N . Notre projecteur est alors $P_{softmax}(\chi) = \beta * \sigma(\chi)$

3.6 Adam

Notre dernière implémentation vise à améliorer la vitesse de convergence ainsi que la qualité de la solution. Pour cela, nous utilisons l'algorithme Adam qui est majoritairement utilisé pour l'optimisation des réseaux de neurones.

Le principe est d'utiliser la notion de moment cinétique, en conservant dans le gradient la vitesse accumulé précédemment, afin d'éviter de tomber dans des minimums locaux. L'implémentation suit le papier original :

```

input :  $\gamma$  (lr),  $\beta_1, \beta_2$  (betas),  $\theta_0$  (params),  $f(\theta)$  (objective)
          $\lambda$  (weight decay), amsgrad
initialize :  $m_0 \leftarrow 0$  ( first moment),  $v_0 \leftarrow 0$  (second moment),  $\widehat{v}_0^{max} \leftarrow 0$ 

```

```

for  $t = 1$  to ... do
     $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ 
    if  $\lambda \neq 0$ 
         $g_t \leftarrow g_t + \lambda \theta_{t-1}$ 
     $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) g_t$ 
     $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ 
     $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ 
     $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ 
    if amsgrad
         $\widehat{v}_t^{max} \leftarrow \max(\widehat{v}_t^{max}, \widehat{v}_t)$ 
         $\theta_t \leftarrow \theta_{t-1} - \gamma \widehat{m}_t / (\sqrt{\widehat{v}_t^{max}} + \epsilon)$ 
    else
         $\theta_t \leftarrow \theta_{t-1} - \gamma \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ 

```

```

return  $\theta_t$ 

```

3.7 Algorithme génétique

En plus des algorithmes de descente de gradient, nous avons implémenté une deuxième méthode d'optimisation : les algorithmes génétiques. Les algorithmes génétiques sont inspirés de la sélection naturelle. Le principe est le suivant:

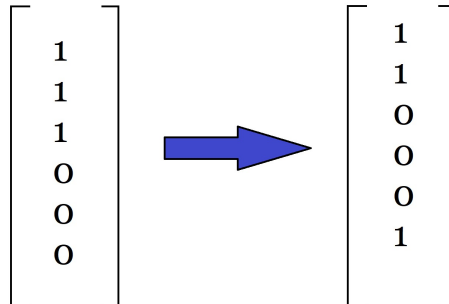
1. Soit une population où chaque individu représente une solution de notre problème d'optimisation, donc ici une matrice de la taille du maillage représentant χ
2. Pour chaque itération de l'algorithme on va :
 - (a) Effectuer des crossovers (simulation de la reproduction de deux individus) : deux individus parents donne un nouvel individu proche de ses parents, les différents crossovers permettent de garantir une convergence de l'algorithme vers un minimum local
 - (b) Effectuer des mutations : les individus de la population vont voir un petit changement dans leur représentation de la solution, les différentes mutations permettent de garantir une certaine exploration de l'ensemble de solution pour espérer trouver le minimum global

3. A l'issue des différents crossovers et mutations on va garder les meilleurs individus enfin de garder une taille de population constante

Mutation

Pour muter un individu, on effectue une permutation des ses composantes sur la bordure.

Mutation



Crossover

Pour effectuer le croisement génétique entre deux individus, on effectue une moyenne pondérée entre les deux parents.

Les individus qui minimisent le mieux l'énergie vont plus se reproduire. On utilise pour cela le principe du tournoi : à chaque crossover, pour choisir les parents, on tire 10% des individus et on choisit les deux meilleurs parmi cette sous population.

Algorithme Génétique :

```
Initialiser mutation_rate
Initialiser M la taille de la population
Initialiser population de M individus

Pour K étapes :
    Evaluer les individus au sens de l'énergie
    Pour M/2 étapes :
        Choisir deux parents par la méthode du tournoi
        Obtenir deux bébés par le crossover entre ces deux parents
        Rajouter les deux bébés a la population
    Pour chaque individu :
        Avec un % de chance mutation_rate :
            Muter l'individu
    Eliminer les individus les plus faibles pour
    revenir à une population de M individus
Renvoyer le meilleur individu au sens de l'énergie
```

4 Résultats des analyses

4.1 Comparaison de l'efficacité des algorithmes

Pour comparer les différents algorithmes proposés nous travaillerons sur les paramètres suivant :

- Taille du maillage : $(N = 20, M = 40)$
- Profondeur de la fractale : 1
- Spacestep : $\frac{1}{20}$
- Wavenumber : $\sqrt{2}$
- $\alpha = 5.51 - 5.75j$

Les algorithmes testés sont les suivants :

1. GradientDescent.evolutive_lr_ProjecteGradientDescent
2. GradientDescent.ProjecteGradientDescent,
3. GradientDescent.evolutive_lr_ProjecteGradientDescent_Adam
4. directGradientDescent.DirectGradientDescent
5. directGradientDescent.DirectGradientDescent_Adam
6. softGD.soft_evolutive_lr_ProjecteGradientDescent
7. softGD.soft_evolutive_lr_ProjecteGradientDescent_Adam
8. SoftGD.soft_ProjecteGradientDescent
9. Algorithme génétique

Ainsi tous les algorithmes dénotés par GradientDescent calculent le gradient en utilisant la dérivée au sens de Frechet, et les algorithmes dénotés par directGradientDescent calculent la dérivée par la limite. Finalement, les algorithmes dénotés par softGD utilisent la fonction Softmax pour projeter la solution après la descente de gradient.

Les algorithmes définis par evolutive_lr font évoluer le learning rate au cours du temps.

On obtient alors les résolutions sur la figure 6.

Le meilleur algorithme se trouve être l'algorithme de DirectGradientDescent c'est à dire la descente de gradient projetée en utilisant le projecteur P_l et en calculant la dérivée par la méthode d'Euler en tenant compte de la projection.

On peut noter également qu'en utilisant la même méthode avec l'algorithme Adam, on obtient la même solution avec une convergence bien plus rapide, néanmoins la descente est légèrement moins stable. On utilisera finalement l'algorithme Adam car la convergence en 5 itérations permet d'affectuer un plus grand nombre de benchmark et plus rapidement.

4.2 Comparaison des différents matériaux

Afin de comparer les performances des différents matériaux qu'il est possible d'utiliser, nous avons tracé l'énergie en fonction du nombre d'onde pour différents matériaux.

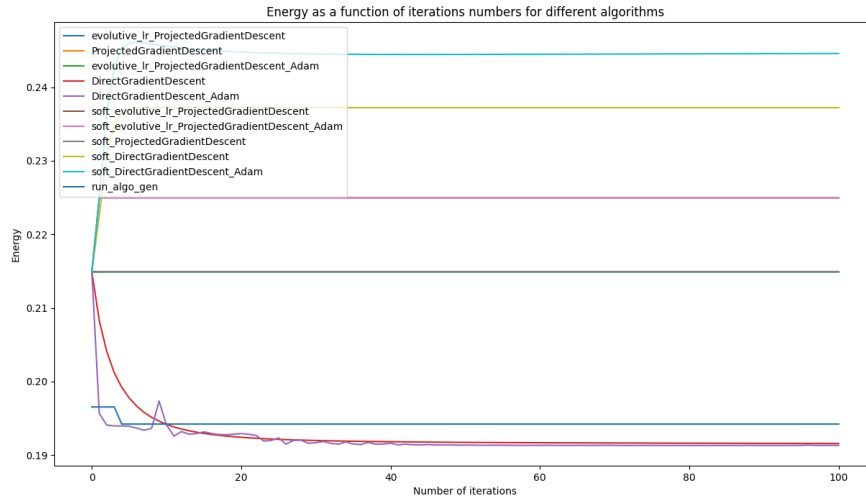


Figure 6: Evolution de l'énergie pour différents algorithmes d'optimisation en fonction du nombre d'itérations

Les matériaux utilisés sont les suivants :

1. Mélamine
2. Sutherland
3. Birchlt
4. Polyurethane

Nous avons tracé ces courbes pour différents niveaux de fractales à savoir sans fractale et avec une fractale de niveau 1.

Nous obtenons alors les différents graphique visible sur les figures 7 et 8 .

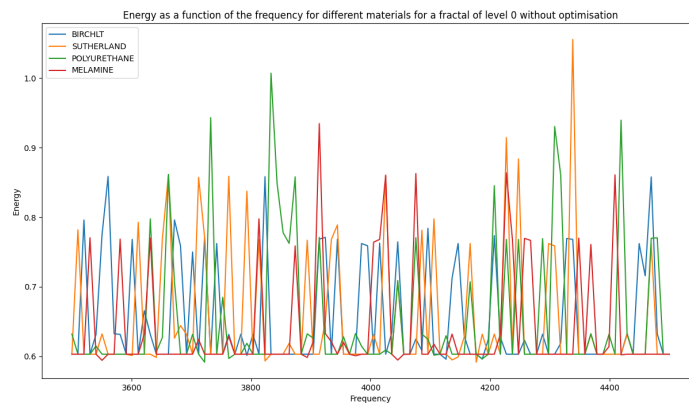


Figure 7: Variation de l'énergie en fonction de la fréquence pour différents matériaux pour un niveau de fractale nul.

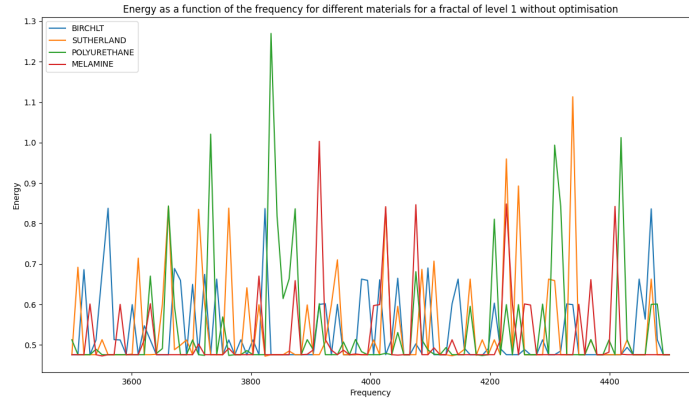


Figure 8: Variation de l'énergie en fonction de la fréquence pour différents matériaux pour un niveau de fractale 1.

Les résultats sont quelques peu brouillés mais il ressort que le sutherland et la mélamine obtiennent les résultats les plus satisfaisants. Pour les départager nous avons retracé l'énergie en fonction de la fréquence pour ces deux matériaux sur une plus grande bande de fréquence après optimisation. On obtient alors les figures 9 et 10 .

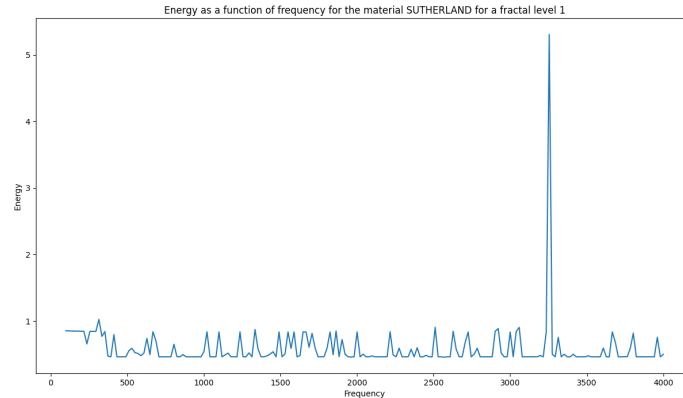


Figure 9: Variation de l'énergie en fonction de la fréquence pour le sutherland

Les deux figures font apparaître une résonance néanmoins le niveau d'énergie de chacune est très différente pour la mélamine on atteint une énergie de $2S.I$ et pour le sutherland une énergie de $5.3S.I$ ce qui conditionne notre choix vers la mélamine. Ce choix sera confirmé par la fréquence de résonance, en effet les fréquences critiques pour un moteur d'avion se situe autour des 3000 Hz et 4000 Hz ce qui élimine le sutherland car sa résonance se situe autour des 3400 Hz.

4.3 Variation de l'énergie en fonction du niveau de la fractale

Maintenant que nous avons trouvé le meilleur algorithme de résolution ainsi que le meilleur matériau, il convient de regarder l'influence de la profondeur de la fractale sur l'énergie.

Nous avons pour cela tracé l'évolution de l'énergie en fonction du niveau de fractale après optimisation. On obtient le graphique suivant à la figure 11.

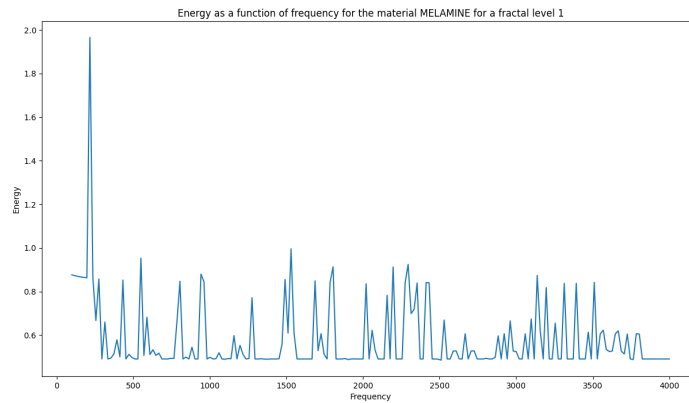


Figure 10: Variation de l'énergie en fonction de la fréquence pour la mélamine.

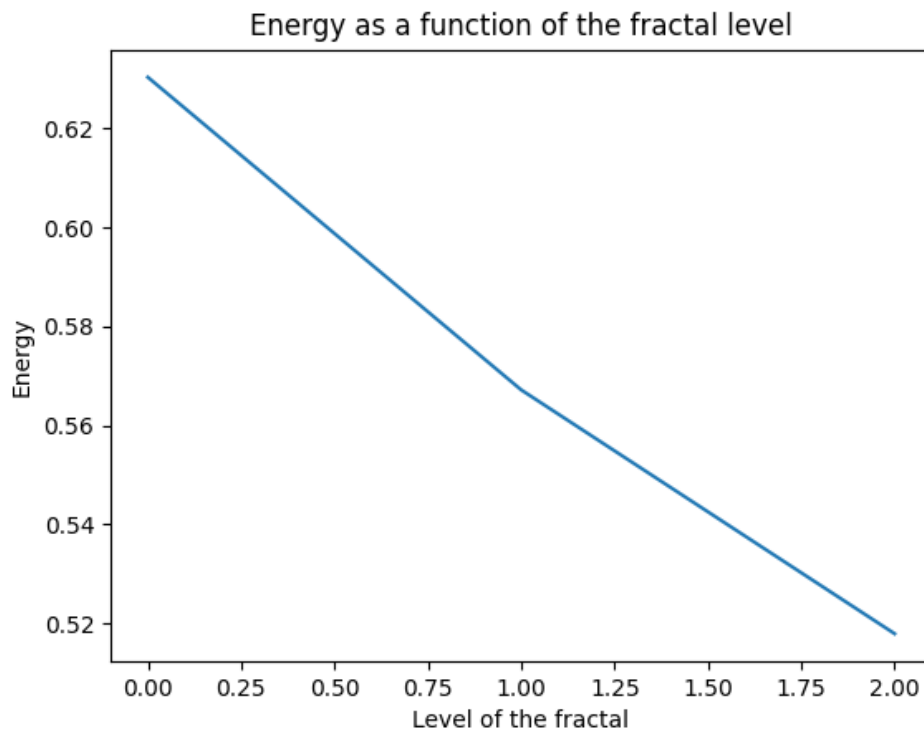


Figure 11: Variation de l'énergie en fonction du niveau de la fractale.

Comme attendu est vu dans le cours, plus le niveau de la fractale augmente et que le nombre d'irrégularité augmente l'énergie est localisé au niveau de la fractale et est donc absorbé par le liner. Ce qui se traduit par une courbe décroissante en fonction du niveau de la fractale.

4.4 Exemple de résultat d'optimisation

Après avoir défini les différents paramètres optimal d'optimisation, nous pouvons lancer des simulations d'optimisation pour voir les différents résultats qu'il est possible d'obtenir.

Dans un premier temps, nous avons optimisé la frontière pour le problème suivant :

1. Une discrétisation donnant une grille de taille $(200 * 100)$
2. Un nombre d'onde $k = \sqrt{2}$
3. On utilise la mélamine comme matériau poreux
4. Une fractale de niveau 3

On est parti d'une frontière initialisée telle que donnée sur la figure 12.

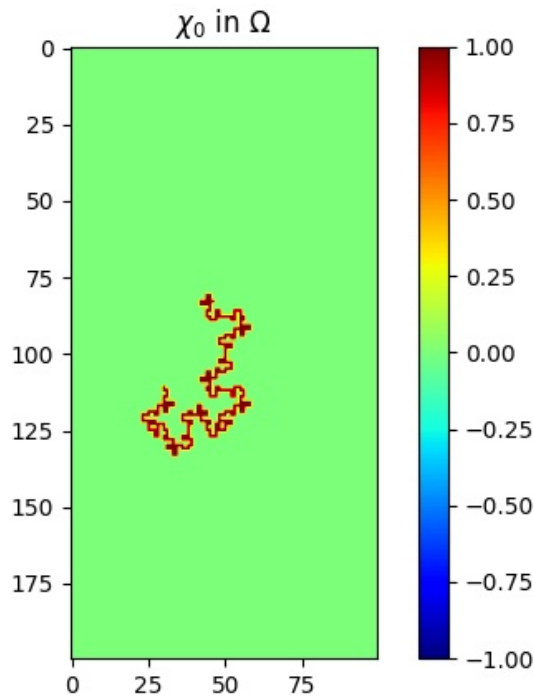


Figure 12: Répartition initiale du matériau poreux

Pour l'optimisation on a utilisé ici l'algorithme de descente de gradient avec calcul du gradient par la limite.

On obtient après 20 itérations la frontière suivante 13

Ce qui correspond une modification de la solution suivante pour la partie réelle 14 et 15 pour la partie imaginaire.

Par la suite on a cherché à optimiser la répartition de matériaux poreux sur la frontière pour une plage de fréquence. On a gardé la plage 0 Hz 4000 Hz car c'est la plage d'intérêt pour un moteur d'avion.

Dans un premier temps on a calculé la frontière optimisée pour le fréquence de résonance observée sur la figure 10. Puis on a calculé l'énergie obtenue avec une telle frontière pour l'ensemble de la plage de fréquence, on obtient la figure suivante 16.

On remarque un gros gain au niveau de la résonance avec une diminution d'énergie de l'ordre de 55%, néanmoins sur le reste de la plage de fréquence le changement de répartition sur la frontière n'a pas une influence positive sur le reste de la plage de fréquence. En effet, on observe une augmentation de l'énergie sur un certain nombre de fréquence.

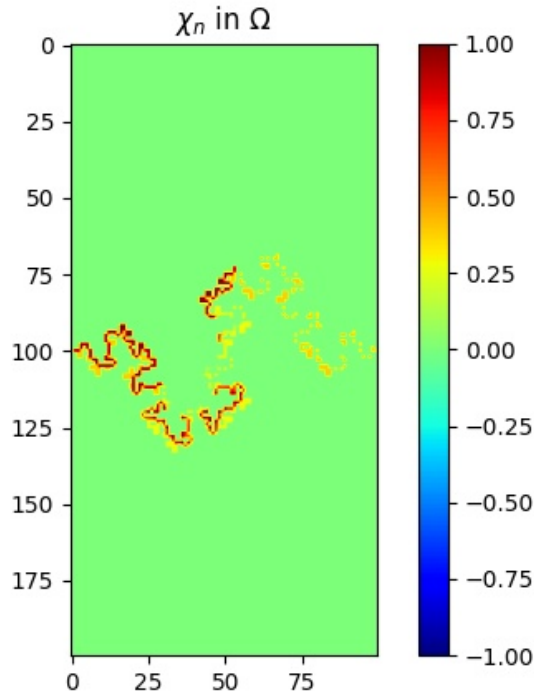


Figure 13: Répartition finale du matériau poreux

Pour corriger cela on a calculé la frontière optimisée pour une fréquence différente on a choisi arbitrairement 2000 Hz qui se situe au centre de la plage de fréquence d'intérêt. On obtient la figure 17.

Cette fois on remarque que l'optimisation a eu un effet sur un ensemble de fréquences néanmoins la résonance a été amplifiée pour palier cela on pourrait effectuer l'optimisation sur la somme de l'énergie à différentes fréquences. Néanmoins cette amélioration n'a pas été effectuée dans le cadre de ces EI mais reste une piste d'amélioration.

4.5 Conclusion

Pour conclure, nous pouvons faire un résumé de ce que nous avons produit au cours de cette semaine. Nous avons implémenté différents algorithmes, et nous avons vu que l'algorithme le plus efficace sur notre problème est la descente de gradient avec un calcul du gradient par la limite.

Nous avons alors déterminé l'énergie en fonction du matériau utilisé, et nous avons alors conclu que la mélamine donnait des résultats optimaux.

Ensuite, nous avons observé un résultat vu dans le cours, l'augmentation des irrégularités sur la frontière permet de localiser l'onde au niveau de celle-ci et donc avec un matériau absorbant diminuer l'énergie de l'onde.

Ces observations nous ont permis de réduire considérablement l'énergie totale de l'onde acoustique, et donc de limiter le bruit produit par le réacteur de l'avion, tout en respectant les contraintes concernant la quantité de liners disponible.

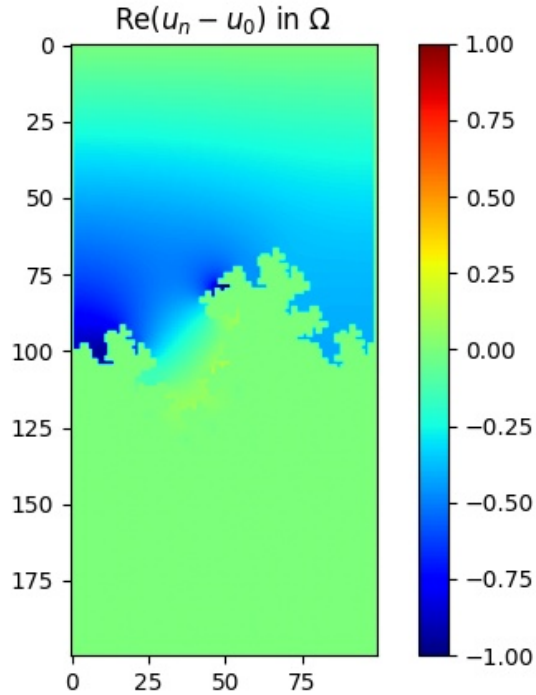


Figure 14: Répartition finale du matériau poreux

5 Proposition d'amélioration

La première amélioration suggérée provient de l'observation faite sur l'optimisation sur une large plage de fréquence et d'implémenter une optimisation qui prend en compte l'énergie sur une plage de fréquence en prenant par exemple comme fonction objective la somme de l'énergie à différence fréquence peut être pondéré pour donner du point aux fréquences critiques.

Ensuite une seconde amélioration possible serait d'accélérer l'exécution du code en particulier la résolution des équations aux dérivées partielles. En effet, l'algorithme le plus performant développé : La descente de gradient par calcul de la limite est très lourd en calcul car pour chaque itération il doit résoudre $O(|Frontiere|)$ fois les équations aux dérivées partielles pour calculer la différence d'énergie de chaque dérivée partielle du gradient.

La dernière amélioration que nous proposons qui suit l'observation que l'initialisation de la frontière χ a une influence importante sur la qualité de l'approximation. On propose donc l'implémentation d'un algorithme mémétique, c'est un algorithme qui s'appuie sur un algorithme génétique mais rajoute de la recherche de minimum local, cela permet de combiner l'algorithme génétique avec la descente de gradient.

References

- [FS21] Éric Savin Frank Simon. Parametric optimization of an acoustic liner. *Edunao*, 14(2), 2021.

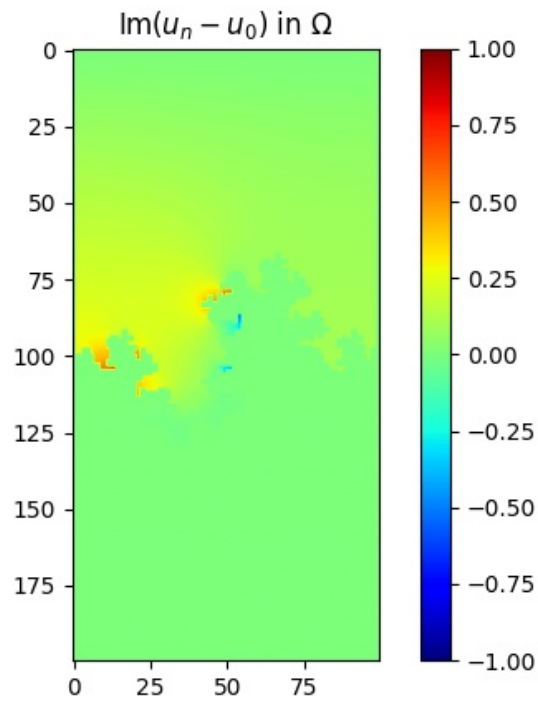


Figure 15: Répartition finale du matériau poreux

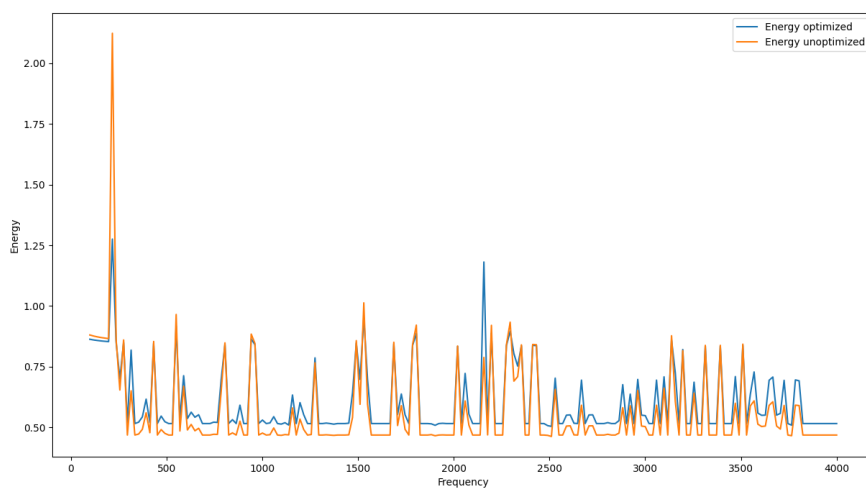


Figure 16: Résultat de l'optimisation sur la résonance

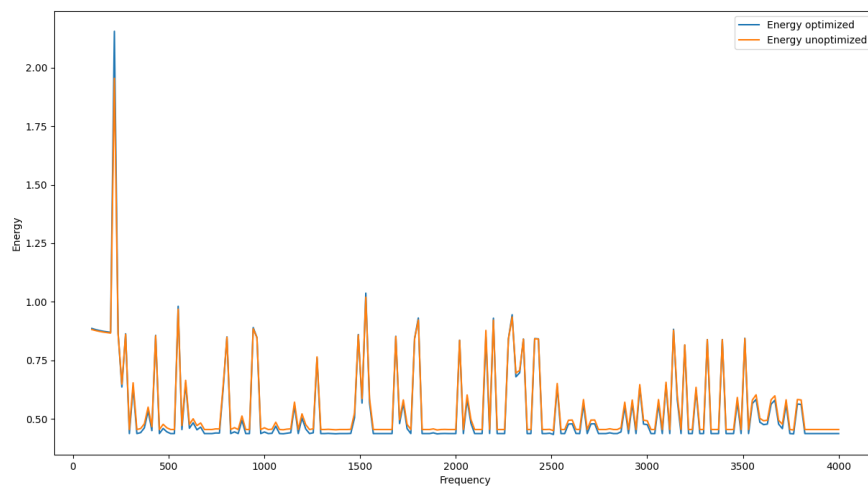


Figure 17: Résultat de l'optimisation à 2000 Hz