Peut-on entendre la forme d'un tambour ?

Ben Achour Elies - Chanut Valentin

Université Paris Saclay

Résumé

Dans cet article, on se propose de développer un point de vue original sur la résolution de l'équation de D'Alembert sur une surface. En s'appuyant sur un protocole développé dans Faure s. d. L'idée étant de partir du spectre de la surface pour décrire le comportement des ondes. Nous mettons en place une résolution numérique avec une programmation orientée objet à la fois pour simplifier l'interaction de l'utilisateur avec le code et d'autre part pour refléter une certaine vision du problème. Nous testons l'efficacité de notre code sur des résultats exacts empruntés à la géométrie spectrale. Nous retrouvons notamment le spectre du tambour rectangulaire et cylindrique avec une relative précision qu'on essaye de quantifier. On retrouve également la loi de Weyl qui prédit la distribution asymptotique des modes. Enfin, on donne un exemple d'une paire de tambours isospectraux, c'est à dire deux géométries différentes partageant les mêmes fréquences de résonance. Cette étude nous permettra de discuter de la validité de l'approche et notamment de la discrétisation du domaine.

1 Introduction

Peut-on entendre la forme d'un tambour ? Cette question, posée par Marc Kac en 1966, s'inscrit dans le champ de la géométrie spectrale en mathématique qui s'intéresse au lien entre spectre de l'opérateur Laplacien et les propriétés géométriques du domaine considéré. On se propose dans ce projet de modéliser les vibrations d'un tambour et d'élargir l'étude d'un problème classique en physique, en utilisant des résultats mathématiques détaillés Section 1.1.1. Dans un premier temps, on s'attachera à déterminer les fonctions et valeurs propres de la nappe 2D (ie. du tambour) selon sa géométrie. On mettra ensuite en oeuvre un méthode pour faire évoluer temporellement n'importe quelle fonction solution de l'équation d'onde.

1.1 Définition du problème

1.1.1 Modélisation et équation d'onde

On note $u(x, y, t)$ l'élongation de la membrane dans la direction verticale $\vec{e_z}$ au point $(x, y) \in \mathcal{D}$, $\mathcal{D}$ étant le domaine du plan (xy) modélisant la membrane vibrante. On note $\partial\mathcal{D}$ ses bords. En supposant l'angle de déformation est petit et en s'affranchissant du poids on peut appliquer la deuxième lois de Newton à un élément de surface élémentaire pour obtenir l'équation d'évolution suivante :

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u \tag{1}$$

Avec $c = \sqrt{\frac{\tau}{\mu}}$, μ et τ étant respectivement la tension et la masse de la membrane par unité de surface. Par ailleurs on a les conditions aux bords suivantes :

$$\forall (x, y) \in \partial\mathcal{D} \ : \ u(x, y, t) = 0 \tag{2}$$

1.1.2 Fonctions propres

Dans le cas 1D la forme du domaine induit une discrétisation des modes propres de vibration. De manière similaire en 2D, la forme du domaine va contraindre la manière dont vibre la nappe. Il est possible de montrer que l'espace de l'ensemble des fonctions à deux variables du domaine $\mathcal{D}$ vérifiant la condition (2) admet une base $(f_j)_{j\in\mathbf{N}}$ qui sont en fait les vecteurs propres de l'operateur $(-\Delta)$ tel que :

$$(-\Delta) f_j = \lambda_j f_j \text{ et } 0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq ... \tag{3}$$

Ainsi il est possible de décomposer toute fonction $u(x, y, t)$ solution de (1) sur cette base :

$$u(x, y, t) = \sum_{j\geq 0} u_j(t) f_j(x, y) \tag{4}$$

où $u_j(t)$ vérifie $\forall j$:

$$u_j(t) = u_j(0) \cos\left(\sqrt{\lambda_j} t\right) - \frac{1}{\sqrt{\lambda_j}} \frac{\partial u_j(0)}{\partial t} \sin\left(\sqrt{\lambda_j} t\right) \tag{5}$$

En effet on a :

$$\begin{aligned} & -\Delta u = -\frac{\partial^2 u}{\partial t^2} \\ \Leftrightarrow \quad & \sum_{j\geq 0} \lambda_j u_j(t) f_j(x, y) = -\sum_{j\geq 0} f_j(x, y) \frac{\partial^2 u_j}{\partial t^2} \\ \Leftrightarrow \quad & \frac{\partial^2 u_j}{\partial t^2} = \lambda_j u_j(t) \end{aligned}$$

Ainsi connaissant les modes propres pour une forme donnée de tambour et les conditions initiales de la membrane correspondant aux fonctions $u(x, y, 0)$ et $\frac{\partial u(x,y,0)}{\partial t}$ et il possible

de connaître la forme de la membrane en tout point et à tout instant.
Formellement, il faut définir un produit scalaire $\langle\cdot|\cdot\rangle$ sur l'espace des fonctions qu'on étudie tel que :

$$\langle u|v\rangle = \int_{\mathcal{D}} dxdy\ u(x,y,t)v(x,y,t) \tag{6}$$

On a alors simplement que :

$$u(x,y,t) = \sum_{j\geq 0} \langle f_j|u\rangle\,(t)\cdot f_j(x,y) \tag{7}$$

1.2 Résultats exacts

Plusieurs résultats analytiques existent pour ce problème, notamment des solutions exacts pour les fonctions propres du laplacien sur des domaines à haute symétrie, ainsi que des résultats généraux sur la distributions des valeurs propres.

1.2.1 Tambours particuliers

Deux solutions analytiques bien connus sont le tambour rectangulaire et le tambour cylindrique, les résolutions complètes sont disponibles en Annexe ? ?.

On a pour le tambour rectangulaire de taille L_x, L_y :

$$\begin{cases} \lambda_j = & \left(\frac{\pi n_x}{L_x}\right)^2 + \left(\frac{\pi n_y}{L_y}\right)^2 \quad ; \quad (n_x,n_y)\in N^{*2} \\ f_j = & \sin\sqrt{\lambda_j}x\cdot\sin\sqrt{\lambda_j}y \end{cases} \tag{8}$$

Pour le cas cylindrique on a besoin de faire intervenir les zéros des fonctions de Bessel de premières espèce d'ordre m, J_m : x_{nm} est le n-iéme zéro de la fonction J_m.

$$\begin{cases} \lambda_j = & \frac{x_{nm}}{R}^2 \\ f_j = & J_m(\sqrt{\lambda_j}r)\cdot\cos(n\theta+\phi) \end{cases} \tag{9}$$

1.2.2 Loi de Weyl

Le Théorème de Weyl est un résultat exact de géométrie spectrale concernant la distribution asymptotique des valeurs propres du laplacien.
Si on définit, $\rho(\lambda_j) = \#\{\lambda_k < \lambda_j\}$, le théorème nous apprend que :

$$\rho(\lambda_j) \underset{\lambda_j\to\inf}{\sim} \frac{S\lambda_j}{4\pi} \tag{10}$$

avec S la surface du domaine $\mathcal{D}$.

1.2.3 Tambours isospectraux

Une des questions que sont amenés à se poser les chercheurs en géométrie spectrale est celle de l'existence de tambours isospectraux. C'est en quelque sorte la contraposée de la question que l'on se pose dans notre sujet. Dans Even et Pawel s. d., on apprend que de tels contours existent et que la première paire historique (le duo flèche/cocotte) sont trouvées autour des années 90. On donne leur représentation et la manière dont on les a approchés dans ce travail.

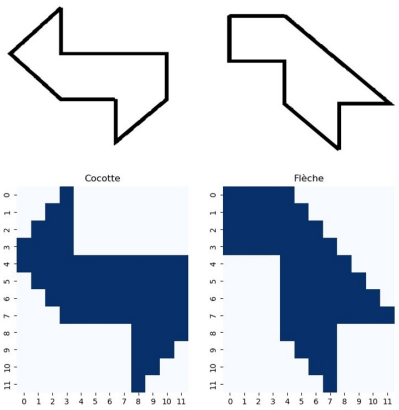


Figure 1 – Contours isospectraux "cocottes" et "flèche" (haut) et contours reproduit avec une précision spatiale plus faible

On se propose dans la section Résultats de comparer les spectres obtenus grâce à notre simulation.

2 Méthode numérique

Pour résoudre ce problème on va utiliser une méthode numérique qui repose sur l'approche spectrale décrite en introduction. Elle consiste donc à déterminer les éléments propres du Laplacien pour une forme donnée de tambour, qui n'existent pas sous forme analytique en général.

2.1 Adimentionnement du problème

On résoud l'équation de d'Alembert (1), où u et c sont dimentionnés respectivement en $[L]$ et $[LT^1]$. Pour simplifier le traitement numérique est pouvoir facilement généraliser la méthode, l'ensemble des grandeurs physiques du problème sont adimensionnées. On choisis comme unité de distance le pas du réseau a. L'unité de temps retenu s'écrit en fonction de a et de c tel que $\tau = \frac{a}{c}$, où c est la célérité des ondes. Ainsi l'unité de temps du problème est le temps que met une onde à se propager sur l'unité de distance.
Dans le code les valeurs de a et c seront choisis égales à l'unité, l'utilisateur pourra donc décrire n'importe quelle situation physique en contractant/dilatant les grandeurs d'espace/temps.

2.2 Représentation de la membrane

On discrétise le plan (xy) en un réseau carré de coté a. Chaque point de l'espace est discret et repéré par deux indices $i = 0,...,N_{11}$ et $j = 0,...,N_{21}$ tel que le domaine $\mathcal{D} \subset \{(x_i;y_j)\ ; x_i = ai,\ y_j = aj\}_{(i,j)\in(0,..,N_{11})\mathrm{x}(0,...,N_{21})}$. On note P le nombre de points qui appartiennent au domaine du tambour.

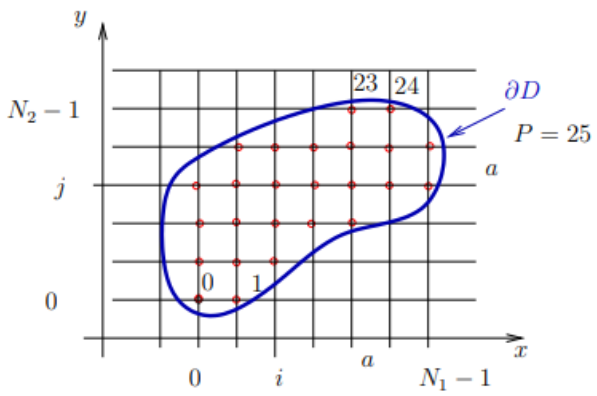


FIGURE 2 – Schématisation de la discrétisation du domaine

On représente une fonction $u(x, y)$ du domaine $\mathcal{D}$ par une matrice $(u_{i,j})_{(i,j)\in(0,..,N_1)^2}$ où $N = \max(N_1, N_2)$ et $u_{i,j} = u(x_i, y_j)$ en supposant que $u_{ij} = 0$ si $(x_i, y_j) \notin \mathcal{D}$.
L'operateur Laplacien peut alors être calculé à l'aide de l'approximation des différences finis :

$$\Delta u(x_i, y_j) \approx \frac{1}{a^2}\left(u_{i+1,j} + u_{i+1,j} + u_{i,j+1} + u_{i,j_1 4} u_{ij}\right) \quad (11)$$

Pour une meilleure représentation des données on va "déplier" la matrice $u_{i,j}$ qui décrit la surface du tambour en un vecteur colonne qui ne contient que les amplitudes des sites à l'intérieur du domaine, $(u_p)_{p\in(0,...,P_1)}$. Il faudra ainsi définir des objets qui permettent ce passage entre représentations :

— $(T_{i,j})_{(i,j)\in(0,..,N_1)^2} : \begin{cases} T_{i,j} = p & \text{si } (i,j) \in \mathcal{D} \\ T_{i,j} = 1 & \text{sinon} \end{cases}$

— $(T_1)_{p\in(0,...,P_1)}$ et $(T_2)_{p\in(0,...,P_1)} : \begin{cases} T_{1p} = i \\ T_{2p} = j \end{cases}$

Ces objets encodent donc explicitement les relations $p(i, j)$ et $\{i, j\}(p)$, nécessaires pour passer d'une représentation à l'autre.

2.3 Discrétisation du Laplacien

Ce choix de représentation ne trouve pas sa principale utilité dans l'ergonomicité de la manipulation des donnés. Il permet en fait de se ramener à une équation aux valeurs propres matricielle pour le laplacien. La matrice $A = (A_{p',p})_{p',p=0,..,P_1}$ représente l'operateur laplacien discrétisé d'après (11) tel que :

$$A_{p,p'} = \begin{cases} \frac{4}{a^2} & \text{si } p = p' \\ \frac{1}{a^2} & \text{si } p \text{ et } p' \text{ sont voisins} \\ 0 & \text{sinon} \end{cases} \quad (12)$$

La diagonalisation numérique de cette matrice, symétrique rélle, donne un ensemble de P vecteurs colonne orthogonaux de taille P, $\{(f_p)_{p\in(0,...,P_1)}\}$ qui représentent chacun la membrane du tambour pour un mode propre. Ainsi que P scalaires, $\{(\lambda_p)\ ;\ p \in (0, ..., P_1)\}$ correspondant au carré des pulsations propres associées à un mode.
On pourra calculer l'évolution d'un tel mode f_p à l'aide de (4), avec $u_j = \delta_{jp}$.

2.4 Description d'une oscillation quelconque

Pour décrire une membrane qui ne correspond pas à un mode propre (u_p), il suffira d'introduire un produit scalaire entre vecteurs qui permet de connaître la décomposition de (u_p) sur l'ensemble des modes propres (f_p) qui constitue alors une base partielle.
Pour calculer l'état de la membrane à un instant t, on fera à nouveau appel à cette décomposition dans (4).

3 Implémentation C++

On met en oeuvre cette méthode, en utilisant le language de programmation C++. On s'attache à développer un code orienté objet, qui assurera d'une part une grande lisibilité et simplicité d'utilisation. D'autre part cette structure de programmation permettra de faire ressortir l'agencement des objets physiques dans notre description théorique.

3.1 Structure de classes

Nous avons défini 4 classes :

1. la classe tambour avec laquelle l'utilisateur interagit
2. les 3 classes geometrie, spectre, nappe.

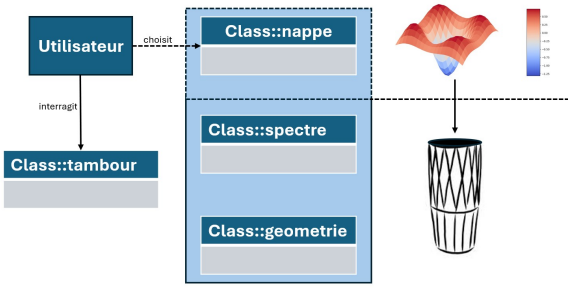


FIGURE 3 – Hiérarchie de la structure de classe

Ce découpage traduit la décomposition théorique du problème, on en donne un schéma plus détaillé figure 4).

— La classe geometrie encode les conditions aux limites imposées par la géométrie d'un contour. C'est aussi au sein de cette classe qu'est géré le processus de "vectorisation" de la surface. Elle a pour attributs 3 entiers N_1, N_2, P qui caractérisent respectivement les dimensions de la matrice carrée sur laquelle on construit le contour et le nombre de points à l'intérieur de la surface. Les autres attributs sont la matrice T et les vecteurs T_1 et T_2 introduits sous-section 2.2. Les méthodes de cette classe contiennent différents constructeurs qui vont nous permettre de définir des contours variés (ellipses,rectangles,contours quartiques), ainsi que toutes les méthodes permettant la traduction des vecteurs contenant les points du domaines à leurs coordonnées sur la matrice T supportant le contour (p_to_ij, voisins(p)...).

— La classe spectre constitue le cœur de ce projet, elle encode le lien entre la géométrie et la distribution de mode. Elle est un intermédiaire entre la géométrie et la membrane réelle. Elle a pour attributs la matrice représentant le laplacien discrétisé (cf. sous-section 2), ainsi que la matrice des vecteurs propres et le vecteur des valeurs propres. Son unique méthode est le constructeur qui à partir d'une geometrie construit la matrice du laplacien discrétisé. Sa construction en dehors de la classe geometrie répond à la fois notre volonté de distinguer un moment clé dans la résolution du problème et à un souci de facilitation d'accès aux vecteurs et valeurs propres.

— La classe nappe représente la membrane réelle du tambour. Elle a pour attributs un vecteur de taille P qui contient les amplitudes des différents points de la membrane, ainsi qu'un vecteur décomposition qui contient la décomposition de la nappe réelle sur la base partielle constituée par le spectre. Les méthodes sont essentiellement des constructeurs variés (nappe à partir d'une décomposition aléatoire, Dirac, ou mode propre).

— Enfin la classe tambour permet à l'utilisateur d'utiliser le code de manière plus fluide et "intuitive", elle a pour attributs, une géométrie et un spectre, et pour méthodes, les constructeurs variés s'appuyant sur ceux de géométrie d'une part, puis des constructeurs s'appuyant sur ceux de nappe d'autre part. Ensuite, elle dispose de toutes les méthodes permettant l'affichage des différents éléments (géométrie, distribution de mode, nappes...). Cette classe joue donc le rôle d'interface utilisateur, ses méthodes entièrement publics possèdent des noms explicites et font ensuite appel aux bonnes methodes/constructeurs des autres classes.

Remarque. On aurait pu définir un unique attribut geometrie pour tambour puis une méthode qui construit le spectre, mais l'appel récurrent au spectre dans les différentes méthodes rendent plus intéressant du point de vue de la complexité temporelle la définition comme attribut.

3.2 Bibliothèque Eigen

Dans ce code, on est amené à diagonaliser la matrice A (cf section Schématisation de la discrétisation du domaine), et à réaliser un certain nombre d'opérations matricielles. Pour cela on fait appel à la bibliothèque Eigen qui propose un ensemble d'outils pour réaliser des opérations en algèbre linéaire. On utilise essentiellement les objets et méthodes suivantes de cette bibliothèque :

— Objets :
  1. Eigen : :VectoxXd, vecteur de doubles (allocation dynamique)
  2. Eigen : :MatrixXd, des matrices de doubles

— Méthodes :
  1. Eigen : :SelfAdjointEigenSolver<Eigen : :MatrixXd>, qui permet de diagonaliser une matrice symmétrique et renvoie une matrice de veteurs propres ainsi qu'un vecteur de valeurs propres, ordonnées dans le sens croissant.

L'ensemble des objets matriciels et vectoriels présentés dans la section 2 sont définis avec la bibliothèque Eigen sauf les deux vecteurs (T_1) et (T_2) qui sont des std : :vector de la bibliothèque standart. En effet, leur taille n'étant pas connue lors de leur définition, on a besoin d'une méthode *pushback()*, indisponible sur Eigen.

3.3 Utilisation du programme

On l'aura compris, l'utilisateur interagit principalement avec la classe tambour, commence par définir un tambour dont on renseigne la géométrie. Il peut ensuite définir différentes nappes sur ce tambour.

3.3.1 Les constructeurs de la classe : :tambour

La classe tambour possède trois constructeurs surchargés. Ils permettent de définir des tambours de différente géométrie :

— Géométrie à symmétrie (xy) :

`tambour(int taille,int type)`

Si type=0, on crée une géometrie carré de côté taille. Si type=1, on crée une géométrie cylindrique de rayon taille.

— Géometrie sans symmétrie (xy) :

`tambour(int tailleX,int tailleY,int type)`

Si type=0, on crée une géometrie rectangle de côté (tailleX,tailleY). Si type=1, on crée une géométrie elliptique de demi-axes (tailleX,tailleY).

— Géometrie quartique :

`tambour(int tailleX,int tailleY,double a, double b`

Définis une géométrie selon l'équation de courbe quartique : $f(x,y) =_1 +4x^4 + 2b(xy)^2 + \frac{y^4}{a}$. On doit donc renseigner les tailles selon x et y ainsi que les coefficients a et b.

3.3.2 Définir une nappe sur un tambour

Une fois le tambour créé, on peut définir un objet nappe sur celui-ci. Pour cela on utilise la méthode de tambour adaptée selon la nappe désirée :

— `new_nappe_with_KG(VectorXd V)` : construit une nappe à partir d'une décomposition sur la base partielle.

— `new_nappe_random()` : construit une nappe avec une décomposition aléatoire.

— `new_nappe_dirac(int i, int j)` : construit une nappe de type Dirac en position (i,j).

— `new_nappe_mode_propre(int k)` : construit le k-ième mode propre du tambour.

— `new_nappe_homogene()` : construit une nappe uniforme.

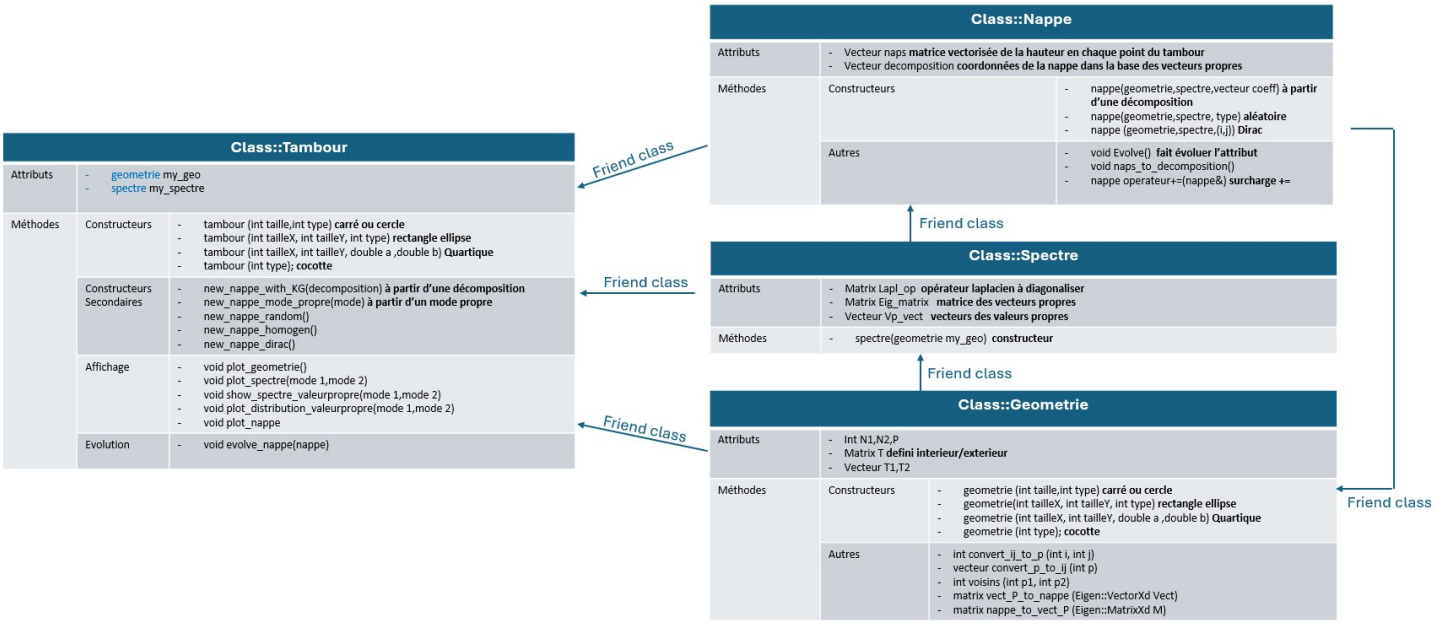


FIGURE 4 – Schéma de la structure de classes

3.4 Exemples

```
int main() {
    tambour my_tamb(10, 13, 0);
    return 0;
}
```

$_2mm$

Crée un tambour rectangulaire de côté 13 par 10.

```
int main() {
    tambour my_tamb(10, 1);
    nappe my_tamb.new_nappe_mode_propre(2) +
    my_tamb.new_nappe_mode_propre(5);
    return 0;
}
```

$_2mm$

Permet de générer une nappe qui est la superposition des modes 3 et 6 sur un tambour circulaire de diamètre 10. Ces deux actions sont la traduction d'un musicien qui choisit son tambour (sa géométrie) puis tape dessus (génère une nappe).
On présente dans la section Résultats certains des usages que l'on peut faire de ce code. Dans la figure 5, on présente les premiers modes obtenus à partir des contours de base que nous pouvons générer ainsi que les lignes de niveaux de ces modes.

4 Résultats

Dans cette section, on cherche à évaluer l'efficacité du code, on y présente la confrontation de nos résultats avec des résultats théoriques.

4.1 Tambours connus

On teste maintenant les performances du programme. On va comparer les résultats analytiques (cf sous-section 1.2.1) aux résultats obtenus par le programme.

4.1.1 Tambour carré

On trace pour des tailles de tambour carré différentes les fréquences propres obtenus avec le programme ainsi que la distribution théorique.

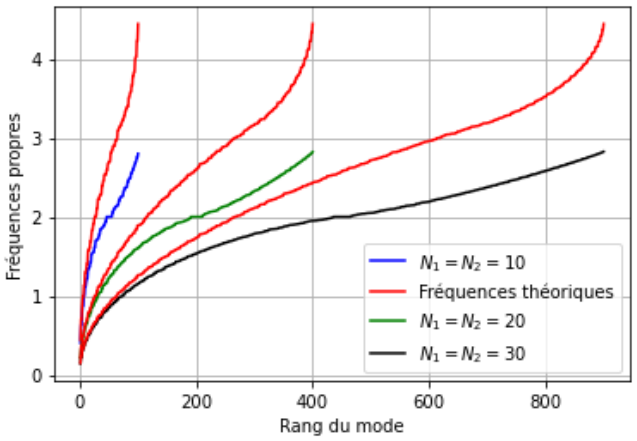


FIGURE 6 – Comparaison des fréquences propres du tambour carré obtenues théoriquement et informatiquement pour plusieurs valeurs de $N_1 = N_2$

On observe Figure 6 que les fréquences premières fréquences propres suivent bien la distribution théorique. Néanmoins assez rapidement les deux distributions se séparent et on obtient des fréquences systématiquement inférieurs à celles attendus. Ceci s'explique par la discrétisation de l'espace, en effet la précision spatialle est limité par le pas du réseau ainsi les hautes fréquences qui correspondent à des petites

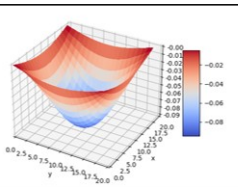
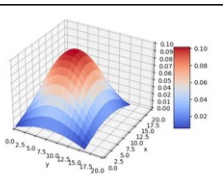
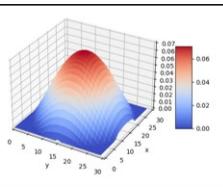
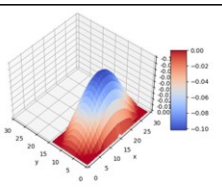
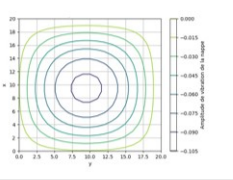
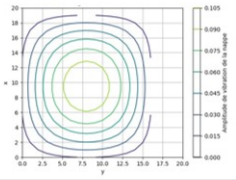
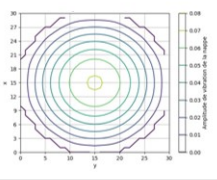
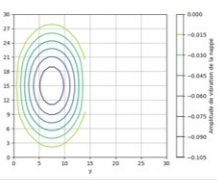
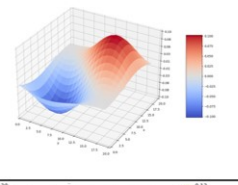
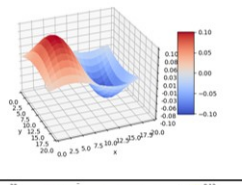
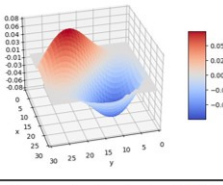
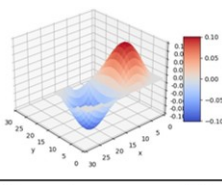
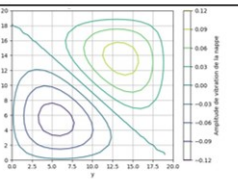
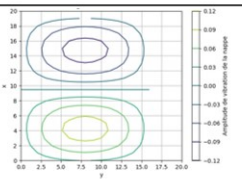
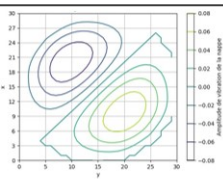
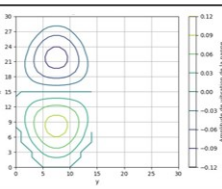
N°mode	Type	Carré	Rectangle	Cylindre	Ellipse
1	nappe				
	Lignes de niveau				
2	nappe				
	Lignes de niveau				

FIGURE 5 – Résumé des premiers modes pour des contours élémentaires

longueures d'ondes sont mal décrites. Pour se convaincre on peut regarder l'évolution de l'écart à la valeur théorique en fonction $N_1 = N_2$. On s'attend à le voir décroitre avec la taille du tambour. On trace donc Figure 7 l'écart relatif pour deux taille de tambour. On constate que cet écart suit une évolution linéaire jusqu'a une certiane valeur seuil.

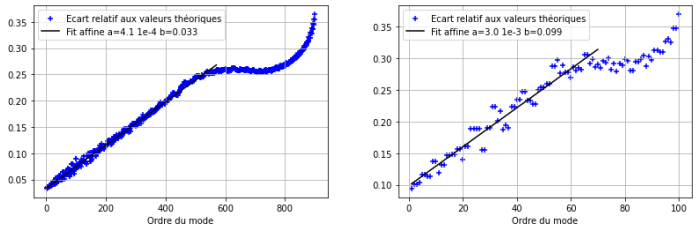


FIGURE 7 – Ecart relatif aux fréquences théoriques pour $N1_1 = N_2 = 30$ à gauche et $N_1 = N_2 = 10$ à droite

On réalise des ajustement affines de ces courbes pour connaitre l'évolution de l'écart avec la taille du tambour. C'est ce qu'on représente Figure **??**, où l'on constate que l'écart relatif suit une loi de puissance en fonction de la taille de la membrane (ie en fait de la précision du pas). Ainsi connaissant cette loi d'évolution on peut choisir à l'avance la taille du tambour selon le nombre de modes qu'on veut obtenir et avec quelle précision.

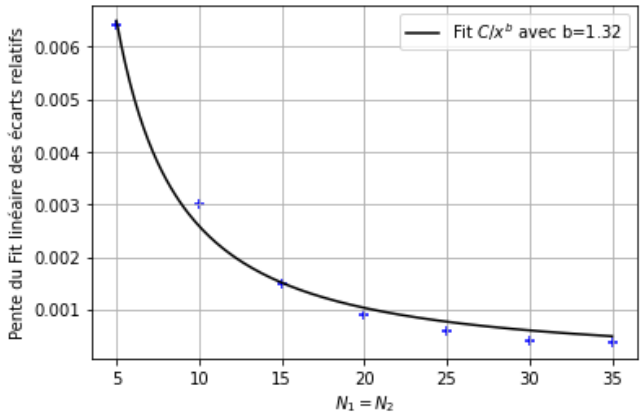


FIGURE 8 – Ajustement avec une loi inverse de la dépendance en $N_1 = N_2$ de l'écart relatif entre les fréquences théoriques et celles obtenues numériquement

4.1.2 Tambour cylindrique

De même pour un tambour a géometrie cylindrique, on compare les résultats numériques et théoriques. On constate Figure 9 que les fréquences divergent beaucoup plus rapidement que ceux obtenus pour la géométrie carré. On retrouve quand même un écart relatif du même ordre de grandeur. Enfin le modes numériques sont toujours inférieurs aux modes théoriques ce qui n'est pas surprenant. Ainsi on voit ici que le programme conserve une certaine performance mais que les modes obtenus sont moins proches que ceux attendus lorsque la géométrie se complique.

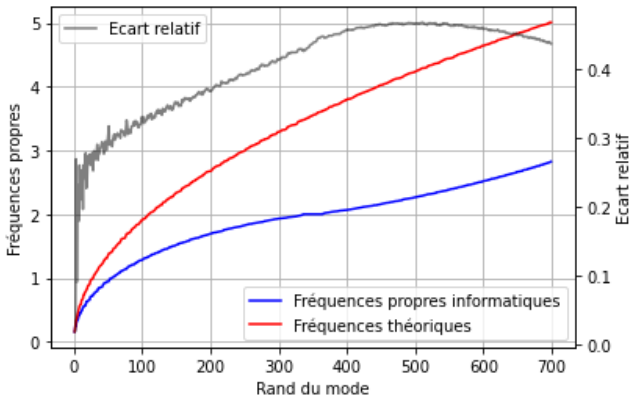
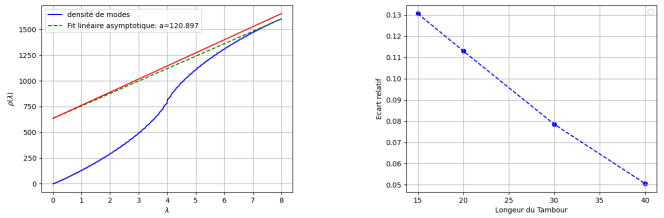


FIGURE 9 – Comparaison des fréquences propres pour un tambour cylindrique de diamètre $N = 30$

4.2 Loi de Weyl

Dans la sous-section Loi de Weyl, on présente un résultat de géométrie spectrale qui porte sur la densité de modes $\rho(\lambda)$. On se propose de vérifier ce théorème dans le cas du tambour carré.



(a) Fit "asymptotique de la distribution de modes pour N=40

(b) Écarts relatifs des coefficients directeurs

FIGURE 10 – Vérification de la loi de Weyl

Pour vérifier la Loi de Weyl, on construit la fonction densité de modes :

$$\rho(\lambda) = \sum_j d\lambda \delta(\lambda - \lambda_j) \tag{13}$$

Et on la fit asymptotiquement avec une loi affine dont le coefficient doit correspondre selon la loi de Weyl à :

$$a_{Weyl} = \frac{S}{4\pi}, \ S = N_1^2 \text{ pour un carré.}$$

On constate que la loi de Weyl est très bien vérifiée. Par ailleurs, plus le "pas" (dilatation du tambour) est petit, plus la loi converge. Ce que l'on peut vérifier figure 10b qui affiche l'écart relatif entre le coefficient de la loi de Weyl et celui qu'on trouve par la simulation.

4.3 Tuiles isospectrales

Dans la sous-section Tambours isospectraux, on a mentionné l'existence de la paire cocotte/flèche. On se propose de retrouver et de comparer leurs spectres(on affiche les fréquences en fonction de leur rang).

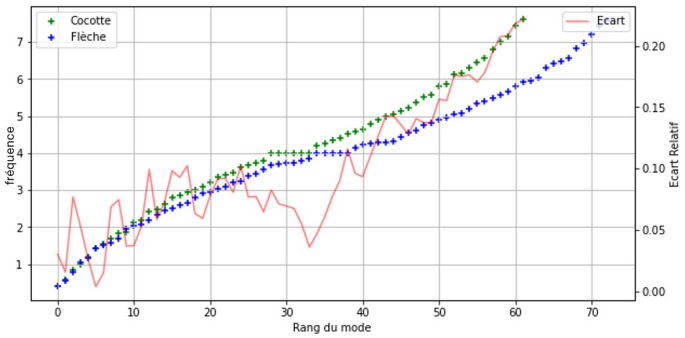


FIGURE 11 – Comparaison des spectres de Cocotte et Flèche on affiche l'écart relatif sur l'axe en face de y)

On constate que les deux spectres sont effectivement très similaires. Malheureusement il s'avère délicat de caractériser la convergence des deux spectres.En effet, étant donné la non-analyticité des domaines faire varier le pas est compliqué.

4.4 Evolution temporelle

Comme introduit dans la sous section Description d'une oscillation quelconque, il est maintenant possible de décomposer et de faire évoluer une onde quelconque sur notre surface. En effet, on utilise la décomposition de l'équation 4 puis on propage cette onde grâce aux modes propres par l'équation 5. Un exemple intéressant est donc obtenu en utilisant une nappe Dirac (à priori partiellement décomposée sur la base des vecteurs propres trouvés).

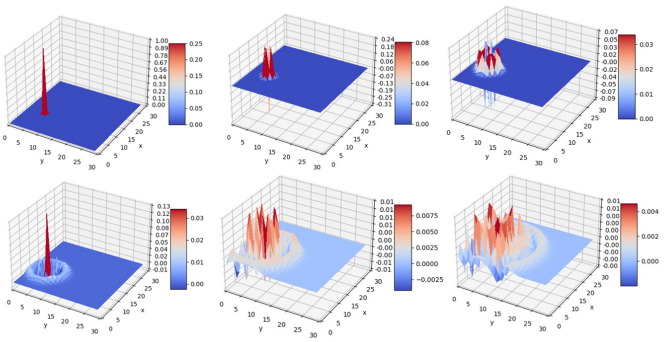


FIGURE 12 – Exemple d'évolution d'un Dirac avec pas de temps arbitraire

5 Conclusion

5.1 Discussions des résultats

Les résultats obtenus sont plutôt satisfaisants et correspondent à nos attentes. Le spectre du tambour rectangulaire est bien prédit, la divergence des résultats de simulation relativement comprise, elle permet également d'approcher la Loi de Weyl. Les tuiles isospectrales donnent aussi un bon résultat avec des spectres très semblables. Par ailleurs, la simulation de l'évolution du Dirac semble correspondre à ce qu'on attendrait de la propagation d'une

onde sans dissipation sur une surface. Si nous devions pousser ce travail plus avant, il nous semblerait intéressant de développer deux directions :

— Caractériser plus précisément la convergence du modèle discret vers les résultats exacts continus. En effet, la variété des résultats abordés permet d'imaginer différents indicateurs (la "vitesse" de convergence des spectres, le taux de dissipation du Dirac qui n'est décomposé que sur une base partielle.)

— Les interactions avec la physique quantique, notamment les différentes allusions au chaos quantiques qu'on a pu lire dans Faure s. d. et Even et Pawel s. d.)

5.2 Expérience personnelle

Sur un plan moins formel ce projet nous a permis de mettre en place une nouvelle structure de programmation permettant de transmettre une certaine philosophie à l'utilisateur, et de définir une hierarchie précise entre les objets manipulés. Ceci s'avère très payant, on hérite en effet d'un code très simple d'utilisation et qui transmet l'intuition physique du problème. La rédaction de cette article nous a permis de faire un pas de plus vers la production de contenu scientifique et de nous confronter à ses attentes en termes de rigueur ou de clarté dans la transmission. En ce qui concerne la documentation, nous avons pris plaisir à chercher des sources de domaines divers (géométrie spectrale, physique de la musique) quitte à harmoniser un peu les notations et la rigueur, mais également sous des formes variés articles, vidéos et une conférence qui nous avait à priori inspiré Anantharaman 2023 pour développer la question initiale en dimension supérieure.

6 Annexe

6.1 Résolutions Analytique

On cherche à résoudre l'EDP suivante, à priori analytiquement insoluble :

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u \tag{14}$$

Cette équation porte sur $u(\vec{r}, t) : \mathbf{R}^2 \times \mathbf{R} \to \mathbf{R}$.

Remarque. Le développement suivant est en partie tiré de Henrot 2016/2017

On se focalise sur la détermination de modes propres ou stationnaires : on écrit $u(\vec{r}, t) = \phi(\vec{r})g(t)$, l'équation devient :

$$\frac{\Delta\phi}{\phi} = \frac{g''}{c^2 g} = \lambda$$

On peut justifier $0 > \lambda$

Ainsi, on réécrit l'équation (14) sur ϕ définie de $\mathcal{D} \to \mathbf{R}$ puis on intégre sur $\mathcal{D}$ après avoir multiplié par ϕ ce qui donne :

$$\Delta\phi = \lambda\phi$$
$$\int_{\mathcal{D}} \phi\Delta\phi dS = \lambda \int_{\mathcal{D}} \phi^2 dS$$

Une intégration par partie donne :

$$\int_{\partial\mathcal{D}} \phi\nabla\phi \cdot \mathbf{n}\, ds = \lambda \int_{\mathcal{D}} \phi^2\, dS$$

Il vient en considérant les conditions de Dirichlet $\phi(\vec{r} \in \partial\mathcal{D}) = 0$: la membrane est fixée au tambour :

$$\boxed{\lambda = -\frac{\int_{\mathcal{D}} \nabla\phi^2 dS}{\int_{\mathcal{D}} \phi^2 dS}} \tag{15}$$

On constate rétrospectivement $\lambda > 0$ et donc $\omega^2 > 0$. Plus généralement on a trouvé l'équation que l'on cherche à résoudre ici les conditions aux limites vont imposer des conditions sur ϕ et donner une distribution de modes (ici les k).

6.1.1 Tambour Rectangulaire

Pour un Tambour rectangulaire, on écrit $\phi(x, y) = f(x)g(y)$, il vient

$$g\partial^2_{xx} f + f\partial^2_{yy} g = -k^2 fg$$
$$\frac{\partial^2 xxf}{f} + \frac{\partial^2_{yy} g}{g} = -k^2$$

Les conditions de Dirichlet imposent :

$$\phi(x, y) = \sin(k_x x)\sin(k_y y)$$
$$\begin{cases} k_x = \dfrac{n_x \pi}{L_x} \\ k_y = \dfrac{n_y \pi}{L_y} \end{cases} \text{ avec } n_x, n_y \in \mathbf{N}$$

Ainsi On trouve la distribution de mode suivantes :

$$\boxed{k_{n_x,n_y} = \pi\sqrt{\frac{n_x^2}{L_x} + \frac{n_y^2}{L_y}}} \tag{16}$$

6.1.2 Tambour cylindrique

Pour un Tambour Cylindrique : $\phi(r, \theta) = f(r)g(\theta)$

$$g\partial^2_{rr} f + g\frac{1}{r}\partial_r f + \frac{1}{r^2} f\partial^2_{\theta\theta} g = -k^2 fg$$
$$\left(r^2 k^2 + \frac{r^2 f''}{f} + \frac{rf'}{f}\right)(r) = -\frac{g''}{g}(\theta) = -k^2$$

Par l'équation 15 on peut écrire $\lambda = -n^2$ avec $n \in \mathbf{R}^+$. Ainsi on déduit avec la 2π périodicité de g :

$$g(\theta) = A\cos(n\theta) + B\sin(n\theta) \quad n \in \mathbf{N}^* \tag{17}$$

Il vient avec le changement de variable : $x = kr$

$$x^2 f'' + xf + \left(x_n^{2\,2}\right) f = 0$$

On reconnait l'équation vérifiée par une fonction de Bessel et d'ordre n (on ne conservera dans la solution que la fonction de première espèce, donc définie en 0). Les solutions sont donc de la forme :

$$f_m(r, \theta) = AJ_m(kr) \cdot \cos(n\theta + \phi)$$

avec $J_m(kr)$ la fonction de Bessel de première espèce.
Les conditions aux limites imposent pour un tambour de rayon R :

$$kR \in \{x \in \mathcal{R} / J_m(x) = 0\}$$

On obtient donc la distribution de modes suivantes

$$\boxed{k_{n,m} = \left\{ \frac{x_n}{R} \in \mathbf{R} \ / \ J_m(x_n) = 0, \ (m,n) \in \mathbf{N} \times \mathbf{N}^* \right\}} \tag{18}$$

x_n correspond au n-ième zéro de la fonction de Bessel d'ordre m.

Références

Anantharaman, Nalini (2023). "Peux-t-on entendre la forme d'une bouteille". In : Institut Henri-Poincaré.

Even, Catherine et Pieranski Pawel (s. d.). *Mathématiques et physique en résonance Entendre la forme d'un tambour*. URL : https://www.lptms.universite-paris-saclay.fr/nicolas_pavloff/files/2019/11/billlards-isospectraux.pdf.

Faure, Frédéric (s. d.). "Simulation de la membrane vibrante d'un tambour". fr. In : ().

Henrot, Antoine (2016/2017). *Equations aux dérivées partielles*.