

GYPsilab est un nouvel environnement de prototypage rapide pour la résolution de problèmes variationnels 3D, issus de formulations type éléments finis (FEM) ou équations intégrales (BEM). Entièrement codé en langage Matlab, et disponible en open-source sous licence GPL 3.0, il affiche des performances suffisantes pour résoudre des problèmes numériques de taille raisonnable ( $< 10^6$  inconnues en FEM et  $10^5$  inconnues en BEM). Pour ce faire, l'ensemble des calculs a été vectorisé, et une librairie de compression hiérarchique a été implémentée avec une algèbre complète (openHmx). Ainsi, un nouveau type de matrice est disponible (H-Matrix), surchargeant les matrices usuelles et leurs opérateurs associés (+, -, \*, /, etc.). De plus, une interface simple permet à l'utilisateur d'écrire des formulations variationnelles "à la freefem++", et de résoudre des problèmes complexes sans rentrer dans l'architecture du logiciel. Aujourd'hui, cet outil résout des problèmes en acoustique, vibro-acoustique, électromagnétisme, mécanique des fluides, etc. En guise d'application FEM, un calcul de valeurs propres du Laplacien sera détaillé pour une géométrie volumique (modes de résonance d'une salle vide). Pour une application BEM, un calcul numérique de Head-Related Transfert Function sera présenté. GYPsilab est disponible en téléchargement libre à l'adresse suivante :

<http://www.cmap.polytechnique.fr/aussal/gypsilab/>

## 1 Introduction

La simulation numérique de problèmes d'acoustique est un enjeu à la fois industriel et académique. En milieu industriel, on l'utilise pour comprendre et évaluer le rayonnement acoustique de structures vibrantes (e.g. dans l'industrie automobile, aérienne, spatiale ou marine) afin d'en limiter les incidences sur la structure ou l'environnement. En milieu académique, on s'intéresse à la propagation d'ondes en milieu complexe - par exemple aléatoire - ou à développer de nouvelles méthodes pour calculer la diffraction acoustique d'objets complexes comme les métamatériaux.

Du point de vue des méthodes utilisées pour la modélisation numérique, on utilise couramment la méthode des éléments finis (FEM pour "Finite Element Method"), ou les méthodes de frontière (BEM pour "Boundary Element Method"). Ces deux méthodes ont pour but de résoudre numériquement l'équation aux dérivées partielles modélisant la propagation des ondes acoustique, mais se distinguent du point de vue du type de problèmes que l'on peut résoudre, ainsi que des difficultés numériques que l'on rencontre lors de leur implémentation :

- En FEM, on utilise des maillages volumiques, souvent 3D, et la discrétisation des opérateurs conduit à des matrices creuses. La taille en mémoire de ces objets est proportionnelle au nombre d'inconnues qui servent à décrire l'évolution de la pression. La méthode est très souple et on peut résoudre des problèmes en temps et en espace. Enfin, les matrices étant creuses, lorsque les problèmes sont de grande taille, on se tourne souvent vers des méthodes itératives d'algèbre linéaire pour la résolution des systèmes linéaires. FREEFEM++ [11] est un exemple de logiciel libre permettant de résoudre de manière très souple des problèmes utilisant ce type de méthodes.
- En BEM, on se concentre sur des problèmes harmoniques, à fréquence fixe, et on discrétise les surfaces des objets diffractants. On a ainsi souvent moins d'inconnues, et leur nombre est proportionnel au carré de la fréquence. Par ailleurs, les matrices qui découlent des discrétisations sont pleines et par conséquent la taille en mémoire est proportionnelle au carré du nombre d'inconnues. On peut utiliser pour la résolution des systèmes linéaires sous-jacents des méthodes directes dont le nombre d'opérations

(directement proportionnel au temps de résolution) évoluera comme le cube du nombre d'inconnues ou, autrement dit, la puissance sixième de la fréquence. Ceci empêche bien souvent l'utilisation de cette méthode pour des problèmes dont la taille dépasse la dizaine de milliers d'inconnues, ce qui en pratique est très contraignant. Des techniques relativement récentes, comme par exemple la méthode multipolaire rapide (FMM pour "Fast Multipole Method" [9]), les matrices hiérarchiques (ou  $\mathcal{H}$ -matrices [10]) ou la SCSD [1–3] permettent de contourner cette difficulté, au prix d'une complexification sensible de l'algorithmique. Il existe très peu d'outils logiciels libres permettant d'utiliser ce type de compression de manière simple et transparente.

Malgré ces différences entre les deux approches, il y a une très grande similarité dans la formulation mathématique des problèmes à résoudre par l'une ou l'autre de ces méthodes. En particulier, on utilise très souvent un formalisme Galerkin pour l'écriture de cette formulation, c'est-à-dire que l'on utilise des fonctions tests pour réécrire le problème après intégration dans le domaine de calcul. Ces outils et concepts mathématiques étant similaires, nous proposons une boîte à outils générique, GYPsilab, qui permet de formaliser le problème que l'on souhaite résoudre et d'utiliser l'une ou l'autre (ou les deux) de ces méthodes. Fortement inspiré par FREEFEM++, GYPsilab est intégralement écrite en MATLAB, et permet de formaliser complètement le problème à résoudre de la façon dont on l'écrit mathématiquement. L'idée d'écrire un logiciel en MATLAB pour résoudre des problèmes en utilisant la FEM n'est pas nouvelle [5, 7, 8, 12, 14], néanmoins, à notre connaissance, GYPsilab est le seul logiciel qui permet de résoudre des problèmes en utilisant la FEM ou la BEM indifféremment, et même d'utiliser la technique des  $\mathcal{H}$ -matrices pour compresser les matrices issues de la BEM<sup>1</sup>. Le spectre d'utilisation de GYPsilab est très large et dépasse largement le cadre de l'acoustique. D'autres domaines comme la mécanique des fluides ou l'électromagnétisme ont aussi été abordés. Néanmoins, nous présentons dans la suite, quelques applications en acoustique qui utilisent respectivement de la FEM et de la BEM. Auparavant, nous donnons des exemples introductifs ainsi que les concepts

1. Citons également BEM++ [15], écrit en C++, qui permet de résoudre des problèmes en utilisant la BEM.

qui nous ont guidés pour le développement de GYPSILAB. Des codes de calcul sont également fournis. Enfin, notons que GYPSILAB permet également d'utiliser des méthodes de lancer de rayons pour résoudre les approximations à haute fréquence des problèmes d'acoustique [4].

## 2 Exemples

### 2.1 Une équation de Laplace

Pour commencer, nous prenons l'exemple de l'équation

$$\begin{cases} -\Delta u + u = f \text{ on } \Omega, \\ \frac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega. \end{cases} \quad (1)$$

dans laquelle  $\Omega$  est un domaine bidimensionnel.

La formulation faible permettant une discrétisation par la méthode des éléments finis, est très classique et s'écrit :

$$\int_{\Omega} \nabla v(x) \cdot \nabla u(x) dx + \int_{\Omega} v(x)u(x) dx = \int_{\Omega} f(x)v(x) dx, \quad \forall v$$

Dans le programme qui suit, nous considérons l'approximation  $P^1$  dans laquelle les fonctions considérées sont affines par morceaux sur un maillage triangulaire du domaine. Nous prenons comme exemple la fonction  $f(x, y, z) = x^2$ .

```
% Library paths
addpath('..../openDom')
addpath('..../openMsh')
addpath('..../openFem')
% Mesh of the disk
N = 1000;
mesh = mshDisk(N,1);
% Integration domain
Omega = dom(mesh,3);
% Finite elements
Vh = fem(mesh, 'P1');
% Matrix and RHS
f = @(X) X(:,1).^2;
K = integral(Omega, grad(Vh), grad(Vh)) ...
    + integral(Omega, Vh, Vh);
F = integral(Omega, Vh, f);
% Solving
uh = K \ F;
% Plot the solution
surf(mesh, Vh, uh);
```

Le listing est relativement simple. Après initialisation on maille le domaine (ici un disque unité) avec 1000 sommets. L'espace d'éléments finis est ensuite construit et la formulation est assemblée. La matrice  $K$  du problème, construite avec le mot clé `integral` est creuse. On construit ensuite le second membre et on résout le système ainsi obtenu. Enfin, on trace la solution ce qui fournit la figure 1.

### 2.2 Modes propres d'un volume

Pour le calcul des modes propres de vibration d'un volume  $\Omega$ , on doit résoudre l'équation aux valeurs propres

$$\begin{cases} -\Delta p = \lambda p \text{ dans } \Omega, \\ p = 0 \text{ sur } \partial\Omega. \end{cases} \quad (2)$$

La forme faible de l'équation s'obtient en multipliant par une fonction test  $v$  et en intégrant par parties. On arrive à

$$\int_{\Omega} \nabla u \cdot \nabla v dx = \lambda \int_{\Omega} uv dx.$$

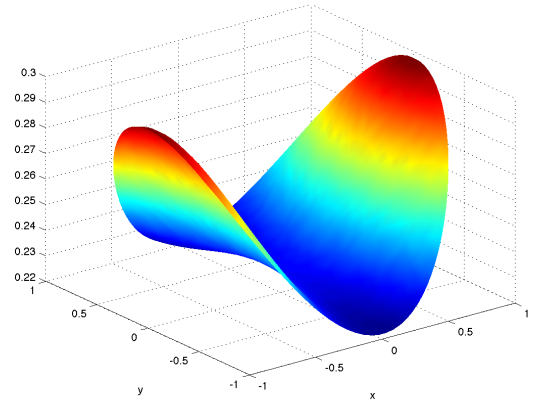


FIGURE 1 – Solution de (1) sur le disque unité en utilisant GYPSILAB.

Après discrétisation par la méthode des éléments finis, il suffit d'assembler les matrices  $A$  de rigidité et  $M$  de masse et résoudre le problème aux valeurs propres généralisé

$$AP = \lambda MP.$$

Dans le listing suivant, nous prenons  $\Omega = [0, 1] \times [0, \frac{1}{2}] \times [0, \frac{1}{2}]$ , que nous maillons avec  $10^4$  points. Nous imposons également les conditions de Dirichlet sur le bord du domaine  $\Omega$ . Enfin, les 10 plus petites valeurs propres (et vecteurs propres) sont calculés.

```
% Library paths
addpath('..../openDom')
addpath('..../openMsh')
addpath('..../openFem')
% Create mesh of the cube + boundary
N = 1e4;
mesh = mshCube(N,[1 0.5 0.5]);
meshb = mesh.bnd;
% Integration domains
Omega = dom(mesh,4);
% Finite element space
Vh = fem(mesh, 'P1');
Vh = dirichlet(Vh, meshb);
% Matrix and RHS
K = integral(Omega, grad(Vh), grad(Vh));
M = integral(Omega, Vh, Vh);
% Resolution
Neig = 10;
[V,EV] = eigs(K,M,Neig,'SM');
```

Le problème s'écrit très simplement comme une transcription en MATLAB des équations mathématiques. Il fournit les résultats donnés dans le tableau 1.

### 2.3 Diffraction acoustique - BEM

Nous donnons maintenant un exemple d'utilisation de la librairie pour des calculs BEM de diffraction acoustique. Nous traitons ci-dessous le cas de la diffraction par une sphère  $S^2$  de 1 m de rayon avec une condition de Dirichlet (sound-soft). Une onde plane incidente, à la fréquence 1000 Hz vient du dessus. La pression générée par l'onde plane est donnée par la formule

$$p_{inc}(x) = \exp(ikx \cdot d) \quad (3)$$

où  $d = (0, 0, -1)$  est la direction de propagation et  $k = \frac{2\pi f}{c}$  est le nombre d'onde, avec  $c = 340 \text{ ms}^{-1}$  la vitesse du son.

TABLEAU 1 – Valeurs propres exactes et approchées de l'opérateur de Laplace avec conditions aux limites de Dirichlet sur le domaine  $[0, 1] \times [0, \frac{1}{2}] \times [0, \frac{1}{2}]$ . Pour chacune des 10 premières valeurs propres, nous donnons les valeurs exacte et approchée fournie par le programme précédent, ainsi que l'erreur relative entre les deux.

Numéro	Valeur exacte	Approximation	Erreur relative
1	88.8264	90.1853	0.0153
2	118.4353	120.5672	0.0180
3	167.7833	171.4698	0.0220
4	207.2617	213.3081	0.0292
5	207.2617	213.4545	0.0299
6	236.8705	243.2719	0.0270
7	236.8705	244.4577	0.0320
8	236.8705	245.0540	0.0345
9	286.2185	296.6374	0.0364
10	286.2185	297.9819	0.0411

On sait que la pression totale  $p_{tot}$  à l'extérieur de la sphère est donnée par la formule  $p_{tot} = p_{inc} + p_{sca}$  dans laquelle la pression diffractée vérifie la formule (voir par exemple [6, 13])

$$p_{sca}(x) = \int_{\mathbb{S}^2} G(x, y) \lambda(y) d\sigma(y). \quad (4)$$

Dans cette formule, le noyau de Green  $G$  est

$$G(x, y) = \frac{e^{ik|x-y|}}{4\pi|x-y|} \quad (5)$$

et la densité  $\lambda$  est calculée par la formule de simple-couche

$$-p_{inc}(x) = \int_{\mathbb{S}^2} G(x, y) \lambda(y) d\sigma(y), \quad (6)$$

pour  $x \in \mathbb{S}^2$ . En résolvant l'équation (6) par la méthode de Galerkin, on doit résoudre

$$\int_{\mathbb{S}^2} \int_{\mathbb{S}^2} \mu(x) G(x, y) \lambda(y) d\sigma(x) d\sigma(y) = - \int_{\mathbb{S}^2} \mu(x) p_{inc}(x) d\sigma(x), \quad (7)$$

dans laquelle la fonction test  $\mu$  et l'inconnue  $\lambda$  sont des fonctions de l'espace discret, ici les fonctions affines par morceaux.

```
% Library path
addpath('.../openMsh')
addpath('.../openDom')
addpath('.../openFem')
addpath('.../openHmx')
% Parameters
N = 1e4; tol = 1e-3;
X0 = [0 0 -1];
% Spherical mesh
```

```
sphere = mshSphere(N, 1);
S2 = dom(sphere, 3);
% Frequency
f = 1000
k = 2*pi*f/340
% Incident wave
PW = @(X) exp(1i*k*X*X0');
% Green kernel: G(x,y) = exp(ik|x-y|)/|x-y|
Gxy = @(X,Y) femGreenKernel(X,Y, '[exp(ikr)/r]', k);
% Finite element space
Vh = fem(sphere, 'P1');
% Operator \int_{Sx} \int_{Sy} psi(x)' G(x,y) psi(y) dx dy
LHS=1/(4*pi)*(integral(S2,S2,Vh,Gxy,Vh));
LHS=LHS+1/(4*pi)*regularize(S2,S2,Vh, '[1/r]', Vh);
% Wave trace --> \int_{Sx} psi(x)' pw(x) dx
RHS = integral(S2,Vh,PW);
% Solve linear system [-S] * lambda = - P0
lambda = LHS \ RHS;
```

Le résultat numérique obtenu par le script précédent est fourni en Figure 2.

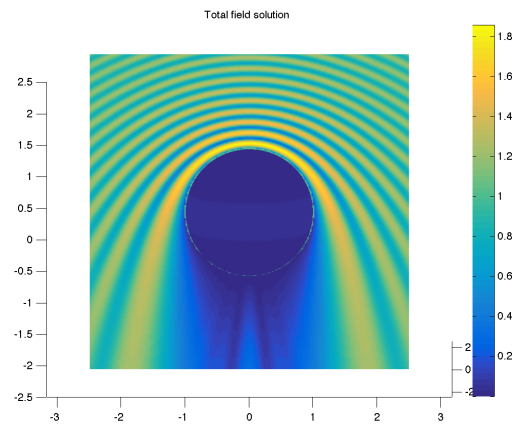


FIGURE 2 – Diffraction acoustique d'une sphere de 1 m de rayon par une onde plane à 1000 Hz provenant du dessus en utilisant GYPSILAB.

### 3 $\mathcal{H}$ -matrices

Lorsque les problèmes résolus en BEM sont de taille trop importante (typiquement supérieure à  $10^4$  inconnues), on ne peut pas stocker la matrice sous-jacente en mémoire. On a alors recours à des techniques de compression de type FMM [9],  $\mathcal{H}$ -matrices [10] ou SCSD [1–3]. GYPSILAB intègre la possibilité d'utiliser de manière transparente la compression en matrices hiérarchiques ( $\mathcal{H}$ -matrices) ce qui permet de pouvoir résoudre des problèmes dont la taille est de l'ordre de  $10^5$  inconnues sur un ordinateur standard. L'utilisation de la librairie OPENHMX se fait de façon transparente au moment de l'assemblage de l'opérateur. Dans le cas de l'acoustique présenté précédemment, il suffit de remplacer la ligne

```
LHS=1/(4*pi)*(integral(S2,S2,Vh,Gxy,Vh));
```

par

```
LHS=1/(4*pi)*(integral(S2,S2,Vh,Gxy,Vh,tol));
```

pour assembler non plus une matrice pleine mais une matrice hiérarchique. Le paramètre `tol` qui est rajouté indique à quelle précision on souhaite construire la matrice hiérarchique. Une valeur typique de `tol=1e-3` s'avère souvent suffisante pour les applications considérées, tout en induisant une compression qui permet des gains en taille significatifs. Par ailleurs, la librairie permet d'utiliser les

$\mathcal{H}$ -matrices de la même façon que l'on utilise les matrices classiques (pleines ou creuses). Les opérateurs classiques (+, -, \*, /) ont été définis sur ce nouveau type, de même que la factorisation LU ou l'affichage (spy). Un exemple de cette dernière fonctionnalité est donné en Figure 4.

A titre d'exemple, nous présentons en Figure 3 le résultat obtenu pour la diffraction d'une onde acoustique par une sphère lorsque le maillage possède 90000 sommets et que l'onde incidente est à 3 kHz.

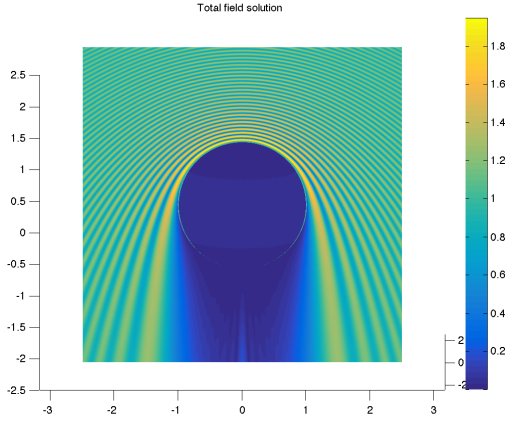


FIGURE 3 – Diffraction acoustique d'une sphere de 1 m de rayon par une onde plane à 3000 Hz provenant du dessus en utilisant GYPSILAB. Le problème est stocké en utilisant les  $\mathcal{H}$ -matrices.

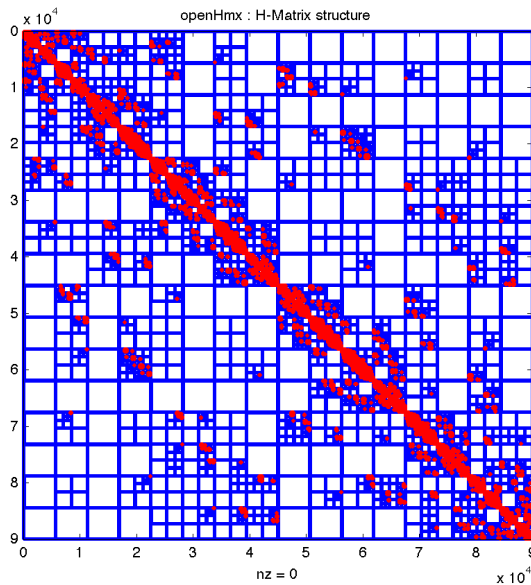


FIGURE 4 – Structure de la matrice hiérarchique dans le problème.

## 4 Applications

### 4.1 Calcul de HRTF

Lors de l'évaluation numérique de HRTF, on doit calculer la diffraction acoustique d'une tête ou d'une tête et d'un torse, à toutes les fréquences audibles. En ce qui

concerne la direction des sources sonores, deux options sont possibles. Ou bien on effectue un calcul pour chaque onde plane provenant de toutes les directions ou bien, en utilisant la formule de réciprocité, on place une source dans le tympan du modèle numérique et on effectue des mesures dans toutes les directions. Nous présentons ci-après deux calculs correspondant à la première situation.

Dans le premier cas, nous considérons un maillage de faible densité d'une tête et calculons la diffraction acoustique pour une onde provenant du dessus de la tête à 8 kHz (voir Figure 5). On remarque le mode qui se forme au sein du pavillon de l'oreille. Ce calcul a été effectué à l'aide de GYPSILAB en utilisant une formulation faisant intervenir l'opérateur hypersingulier (on résout ici un problème de Neumann). Le maillage, de taille modéré, possède 8565 sommets et 17126 triangles. La méthode BEM permet d'effectuer ce calcul en stockant la matrice pleine du problème, sur un ordinateur portable courant.

Le deuxième calcul que nous montrons correspond au même problème, où l'on considère maintenant un domaine représentant la tête et le torse du sujet. Nous souhaitons aussi augmenter la fréquence et passons maintenant à une onde plane incidente à 16 kHz. Le résultat, obtenu grâce à GYPSILAB est proposé Figure 6. Le maillage est beaucoup plus fin et possède maintenant 101115 sommets et 202226 éléments triangulaires. Ce calcul ne pourrait pas passer en mémoire avec une méthode BEM classique sur un ordinateur standard (la matrice seule du problème occuperait plus de 160 Go). Néanmoins, nous avons pu réaliser ce calcul en utilisant les  $\mathcal{H}$ -matrices qui ont permis de compresser la matrice du problème tout en permettant d'en réaliser une factorisation LU (ceci serait particulièrement effectif si l'on doit calculer, par cette méthode, la solution pour toutes les incidences de l'onde plane).

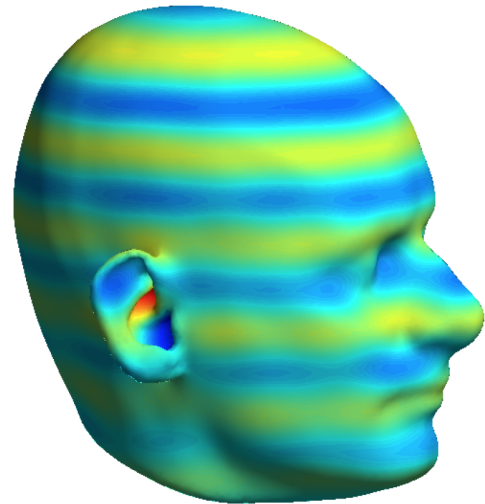


FIGURE 5 – Diffraction acoustique d'une tête par une onde provenant du dessus à 8 kHz.

### 4.2 Acoustique sous-marine

Nous terminons cette rapide revue par un cas de calcul plus industriel, le calcul de la diffraction acoustique d'un sous-marin. La formulation utilisée est la même que



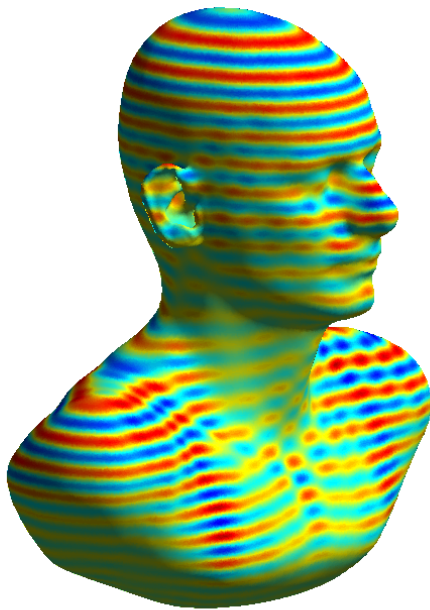


FIGURE 6 – Diffraction acoustique d’une tête et d’un torse par une onde provenant du dessus à 16 kHz.

précédemment, mais l’objet est très différent. Il s’agit d’un sous-marin sur lequel on envoie une onde sonar depuis l’avant. La Figure 7 montre le résultat du calcul et l’on peut y voir non seulement la pression totale sur le bord du sous marin, mais aussi une carte de la pression dans un plan horizontal contenant le sous-marin. Pour ce calcul, nous avons utilisé une formulation du problème avec des conditions de Neuman (obstacle “sound-hard”). La fréquence d’incidence est de 200 Hz et le maillage possède 100 000 sommets, ce qui correspond à autant d’inconnues dans le problème final. Le Tableau 2 donne les temps des différentes opérations sur une machine à 8 cœurs.

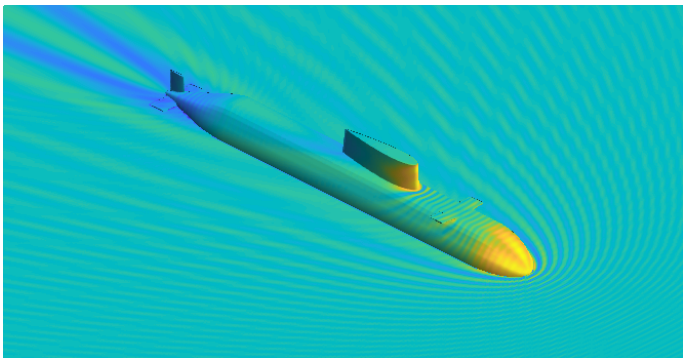


FIGURE 7 – Diffraction acoustique d’un sous-marin par une onde provenant de face à 200 Hz.

TABLEAU 2 – Temps de chacune des opérations pour pouvoir réaliser le calcul sur le sous-marin.

Opération	Temps (s)
Construction de la $\mathcal{H}$ -matrice	1000
Régularisation	101
Factorisation LU	604
Résolution	2.22
Radiation et affichage	1500

## 5 Conclusion

Nous avons présenté GYPSILAB, un nouvel environnement de programmation en MATLAB qui permet la résolution de problèmes d’acoustique. Plusieurs exemples ont été montrés qui permettent d’appréhender le potentiel du logiciel. En particulier, il est particulièrement adapté pour la simulation par FEM ou BEM. GYPSILAB permet de traiter des problèmes dont la taille peut aller jusqu’à  $10^6$  inconnues en FEM et  $10^5$  en BEM sur un ordinateur portable. En particulier, il contient une librairie qui permet d’utiliser le formalisme  $\mathcal{H}$ -matrices pour le stockage et la résolution des problèmes sous-jacents. Ce formalisme, outre qu’il permet d’aborder des problèmes de grande taille, a l’avantage de fournir une algèbre complète matricielle, où l’on peut factoriser les matrices une fois pour toutes. Ceci permet par exemple de résoudre des problèmes possédant de nombreux second-membres beaucoup plus rapidement qu’avec une méthode de type FMM par exemple, dans laquelle la matrice n’est jamais inversée, et le système est résolu par une méthode itérative. Enfin, nous espérons, dans ce rapide tour d’horizon, avoir montré que GYPSILAB s’adresse aussi bien à des utilisateurs académiques qu’industriels.

Enfin, GYPSILAB est disponible<sup>2</sup> en open source sous licence GPL 3.0.

## Références

- [1] Alouges, F. et Aussal, M. : The Sparse Cardinal Sine Decomposition and its Application to fast numerical convolution. *Numerical Algorithms*, 70(2), 427–448 (2015).
- [2] Alouges, F., Aussal, M., Lefebvre-Lepot, A., Pigeonneau F. et Sellier, A. : Application of the sparse cardinal sine decomposition applied to 3D Stokes flows, *International Journal of Comp. Meth. and Exp. Meas.* 5(3) (2017).
- [3] Alouges, F., Aussal M. et Parolin, E. : FEM-BEM coupling for electromagnetism with the Sparse Cardinal Sine Decomposition, accepted to *ESAIM Procs* (2017).

2. Voir <http://www.cmap.polytechnique.fr/aussal/gypsilab/>, ou bien sur GitHub : <https://github.com/matthieuaussal/gypsilab>.

- [4] Aussal, M. et Gueguen, R. : GYPSILAB, un nouvel outil d'auralisation par lancers de rayons, dans ce volume.
- [5] Anjam, I. et Valdman, J. : Fast MATLAB assembly of FEM matrices in 2D and 3D : Edge elements. Applied Mathematics and Computation. 267. (2014). doi :10.1016/j.amc.2015.03.105.
- [6] Colton, D. et Kress, R. : Inverse Acoustic and Electromagnetic Scattering Theory, Second Edition. Springer-Verlag, New York, (1998)
- [7] Cuvelier, F., Japhet, C. et Scarella, G. : An efficient way to assemble finite element matrices in vector languages
- [8] Funken, S., Praetorius, D. et Wissgott, P. : Efficient implementation of adaptive P1-FEM in Matlab, Comput. Methods Appl. Math., 11(4), 460–490 (2011).
- [9] Greengard, L. : The rapid evaluation of potential fields in particle systems. MIT Press (1988)
- [10] Hackbusch, W. : Hierarchische Matrizen. Springer (2009) .
- [11] Hecht, F. : New development in FreeFem++, J. Numer. Math., 20, 3-4, 251–365 (2012). See also <http://www.freefem.org>.
- [12] Kwon, Y. W. et Bang, H. : The finite element method using Matlab, second edition, CRC Press, (2000).
- [13] Nédélec, J.-C. : Acoustic and Electromagnetic Equations, Integral Representations for Harmonic Problems, Springer, 2001.
- [14] Rahman, T. et Valdman, J. : Fast MATLAB assembly of FEM matrices in 2D and 3D : Nodal elements, Appl. Math. Comput. 219, 7151–7158 (2013).
- [15] Śmigaj, W., Arridge, S., Betcke, T., Phillips, J. et Schweiger, M., Solving Boundary Integral Problems with BEM++, ACM Trans. Math. Software 41, 6 :1–6 :40 (2015).