FEM-BEM coupling using GYPSILAB

Matthieu Aussal
joint work with François Alouges

Centre de Mathématiques Appliquées de l'École Polytechnique
Route de Saclay - 91128 Palaiseau CEDEX France

IABEM - Paris
Wednesday 27 June 2018

But what is GYPSILAB ?

GYPSILAB is an **open-source** framework for MATLAB, freely available under GPL 3.0, composed of object-oriented libraries :

- **openMsh :** Multi-dimensional mesh management.
- **openDom :** Numerical integration and variational calculation, *joint work with François Alouges*.
- **openFem :** Finite Element method for 2D and 3D shapes, *joint work with François Alouges*.
- **openHmx :** $\mathcal{H}$-Matrix compression and algebra.
- **openRay :** Accelerated ray-tracing for HF problems, *joint work with Robin Gueguen*.

Designed to combine industrial requirements with fast prototyping.

Matrix formulation for FEM/BEM with GYPSILAB

- **FEM** : local operators $\rightarrow$ **sparse** matrix :

$$[M_{fem}]_{i,j} = \int_{\Omega} \phi_i(\mathbf{x}) f(\mathbf{x}) \phi_j(\mathbf{x}) d\Omega_x.$$

- **BEM** : global operators $\rightarrow$ **full** matrix :

$$[M_{bem}]_{i,j} = \int_{\Gamma_x} \int_{\Gamma_y} \phi_i(\mathbf{x}) f(\mathbf{x}, \mathbf{y}) \phi_j(\mathbf{y}) d\Gamma_x d\Gamma_y.$$

- **Regularization :** Local correction for singular kernel ($|\mathbf{x} - \mathbf{y}| < \epsilon$), semi-analytical integration $\rightarrow$ **sparse** matrix :

$$[M_{reg}]_{i,j} = \int_{\Gamma_x} \int_{\Gamma_y} \phi_i(\mathbf{x}) f(\mathbf{x}, \mathbf{y}) 1_{|\mathbf{x}-\mathbf{y}|<\epsilon} \phi_j(\mathbf{y}) d\Gamma_x d\Gamma_y.$$

And all this matrices are representable using sparse, full or $\mathcal{H}$-Matrix format !

But what is $\mathcal{H}$-Matrix ?

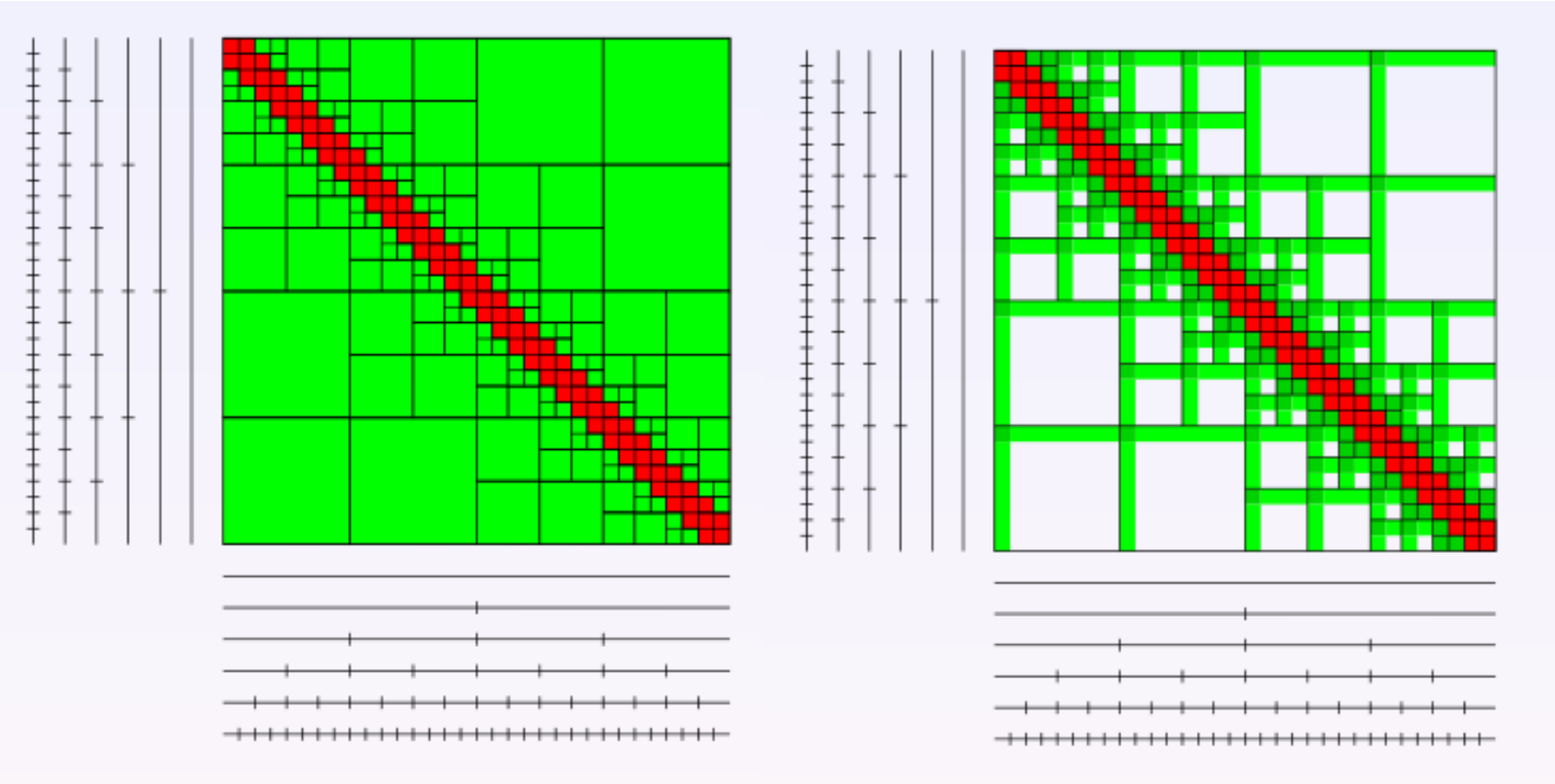
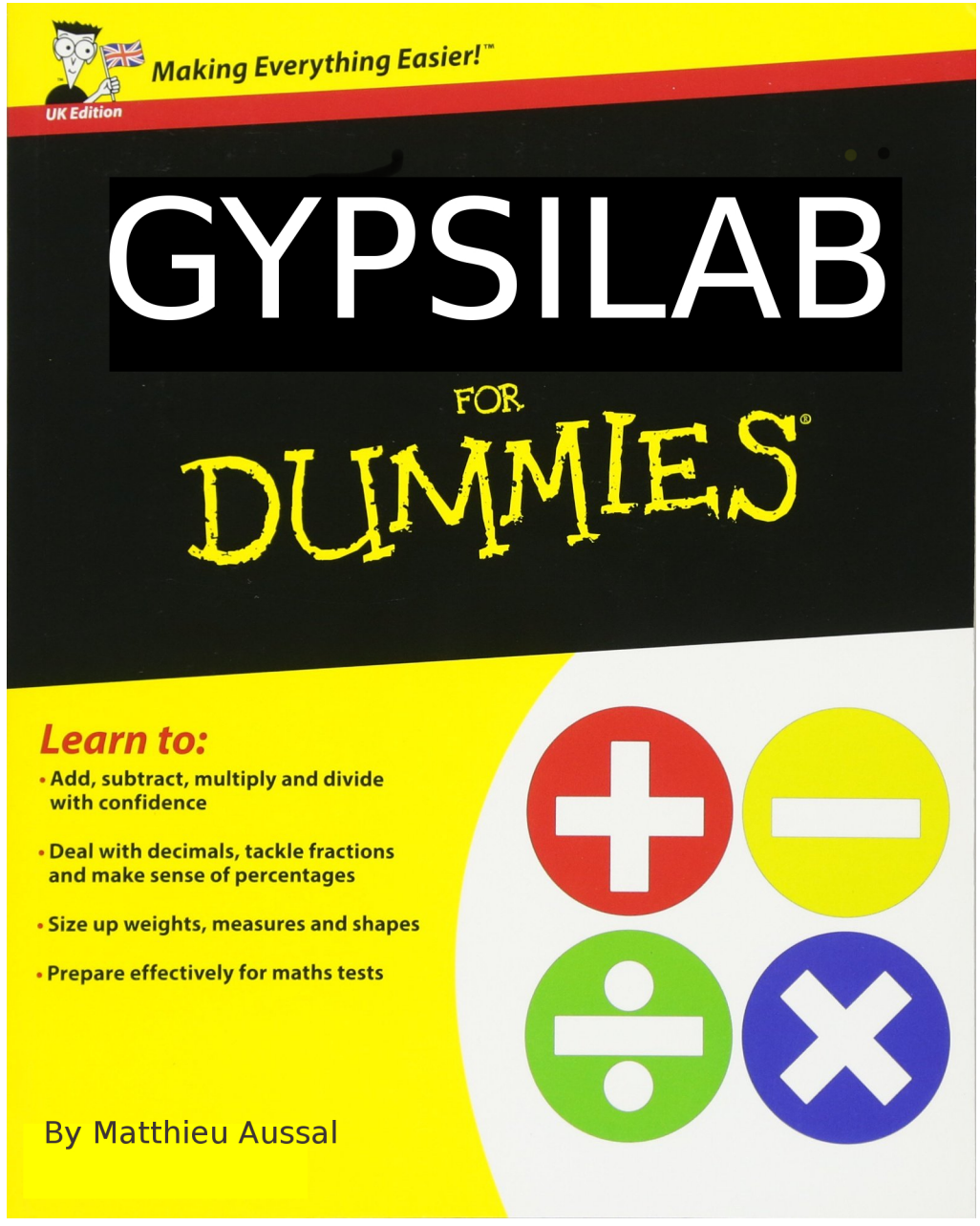


FIGURE : Hierarchical compression, pictures from S. Borm.

More details yesterday, today, and probably tomorrow talks.


Making Everything Easier!™
UK Edition
GYPSILAB
FOR
DUMMIES®
Learn to:
• Add, subtract, multiply and divide with confidence
• Deal with decimals, tackle fractions and make sense of percentages
• Size up weights, measures and shapes
• Prepare effectively for maths tests
By Matthieu Aussal

Example 0 : Discrete convolution with Laplace kernel

Context : For all particles $(\mathbf{x}_i)_{i\in[1,N]}$ and $(\mathbf{y}_j)_{j\in[1,N]}$ in $\mathbb{R}^3$:

$$\begin{aligned} u(\mathbf{x}_i) &= \sum_{j=1}^{N} \frac{1}{4\pi|\mathbf{x}_i - \mathbf{y}_j|} v(\mathbf{y}_j), \\ \Leftrightarrow U &= M * V. \end{aligned}$$

with $M \in M^N(\mathbb{R})$, U and $V \in \mathbb{R}^N$.

Test case : $(\mathbf{x}_i)_{i\in[1,N]}$ and $(\mathbf{y}_j)_{j\in[1,N]}$ are uniformly distributed on the unit sphere S^2.

Computer : 1 core at 2.7 GHz, 32 GO de ram, Matlab R2014a.

Remarks : There is no finite elements in this test case, only evaluate discrete convolution on 3D particles.

Example 0 : Discrete convolution with Laplace kernel

```
% Library path
addpath('../openMsh')
addpath('../openHmx')

% Mesh of a unit sphere
N    = 1000
mesh = mshSphere(N,1);

% Particles data
X = mesh.vtx; Y = X;
V = -1 + 2*rand(N,1);

% Accuracy
tol = 1e-3;

% Laplace green kernel
green = @(X,Y) 1./(4*pi*
    dist(X,Y))
```

```
% Full product
MV = zeros(N,1);
for i = 1:N
    MV(i) = green(X(i,:),
        Y).' * V;
end

% Hierarchical Matrix
Mh  = hmx(X,Y,green,tol);
MhV = Mh * V;

% LU Factorization
[Lh,Uh] = lu(Mh);
LhUhV   = Lh * (Uh * V);

% Plot structure
spy(Mh)
```

Example 0 : Discrete convolution with Laplace kernel

N	Time (s)	Memory peak
10^4	2.04	1 Mo
10^5	328	10 Mo

Full product (quadratic)

N	Build time (s)	MV time (s)	LU time (s)	Mem
10^4	8.24	0.16	9.88	100 Mo
10^5	78.9	1.68	145	1 Go
10^6	1069	25	4120	10 Go

$\mathcal{H}$-**Matrix product and algebra with** $\epsilon = 10^{-3}$ (quasi-linear)

Example 0 : Discrete convolution with Laplace kernel

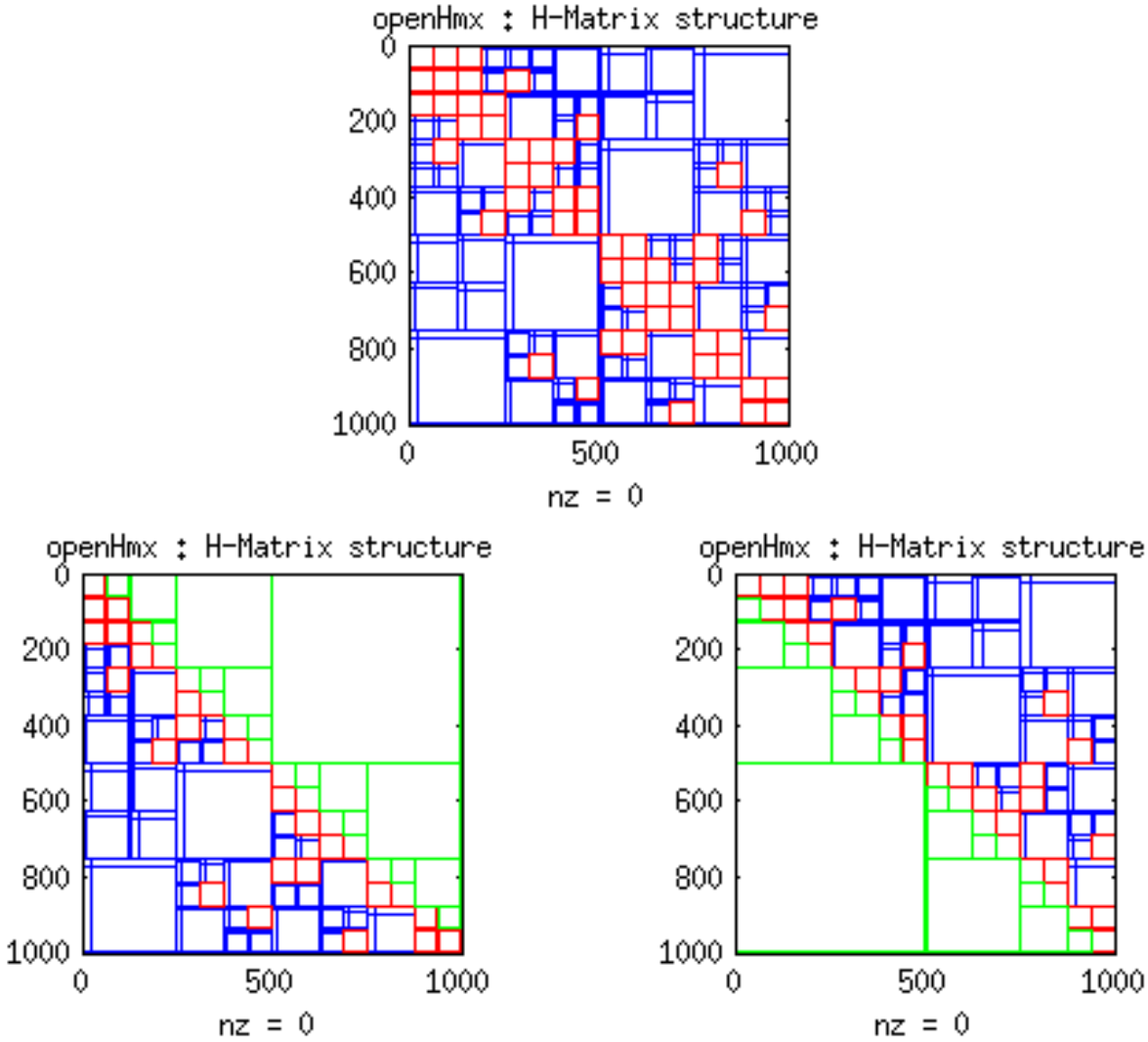


FIGURE : Hierarchical structure of $\mathcal{H}$-Matrix (*spy*)

Example 1 : Laplace Equation

- Laplace equation for all domain Ω :

$$\begin{cases} -\Delta u + u = f \text{ on } \Omega\,, \\ \dfrac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega\,. \end{cases} \tag{1}$$

- Galerkin formulation with finite element discretisation :

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

Example 1 : Laplace Equation

```
% 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
u = fem(mesh,'P1');
v = fem(mesh,'P1');

% Matrix and RHS
f = @(X) X(:,1).^2;
K = integral(omega, ...
grad(v),grad(u))
M = integral(omega,v,u);
F = integral(omega,v,f);

% Solving
uh = (K+M)\F;

% Plot the solution
surf(mesh,u,uh);
```

Example 1 : Laplace Equation

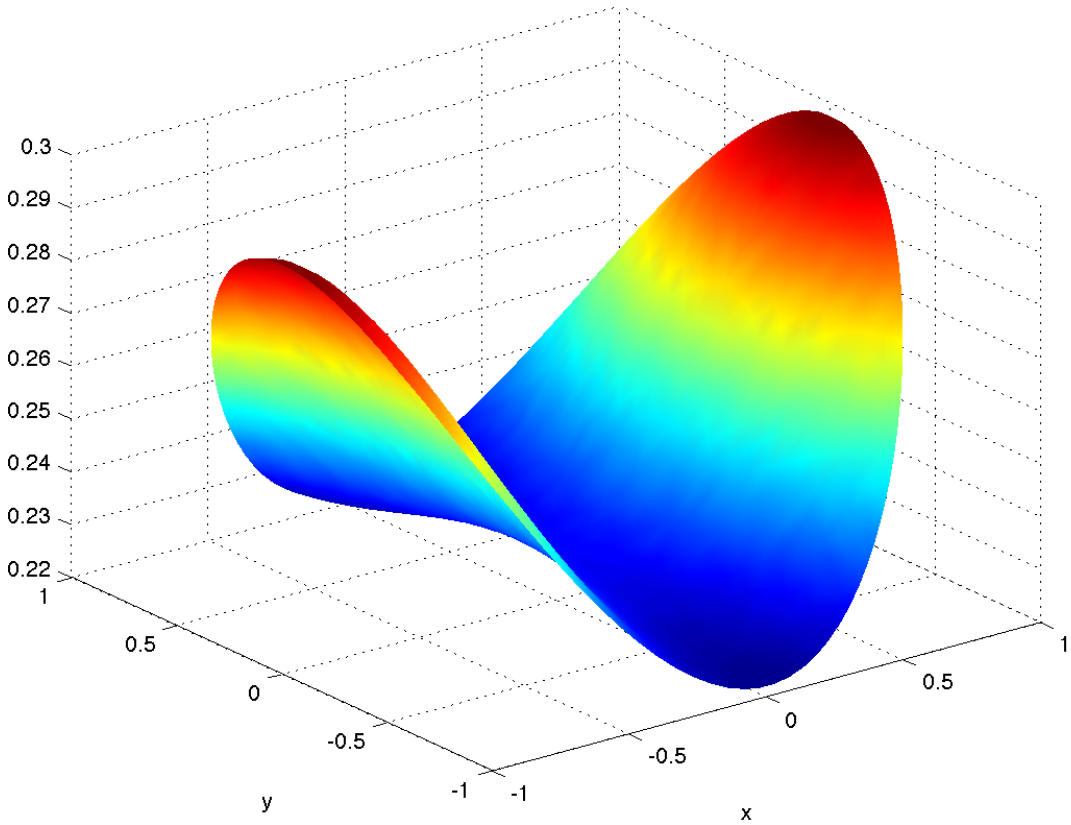


FIGURE : Solution of (1) on the unitary disk $\Rightarrow$ **a chips !**

Example 2 : Volume eigenvalues

- Eigenvalue equation for the volumic Laplacian in Ω :

$$\begin{cases} -\Delta p = \lambda p \text{ in } \Omega\,, \\ p = 0 \text{ on } \partial\Omega\,. \end{cases} \tag{2}$$

- Galerkin formulation with finite element :

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

Example 2 : Volume eigenvalues

```
% Library paths
addpath('../openDom')
addpath('../openMsh')
addpath('../openFem')

% Create meshes
N      = 10^4;
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');
```

Example 2 : Volume eigenvalues

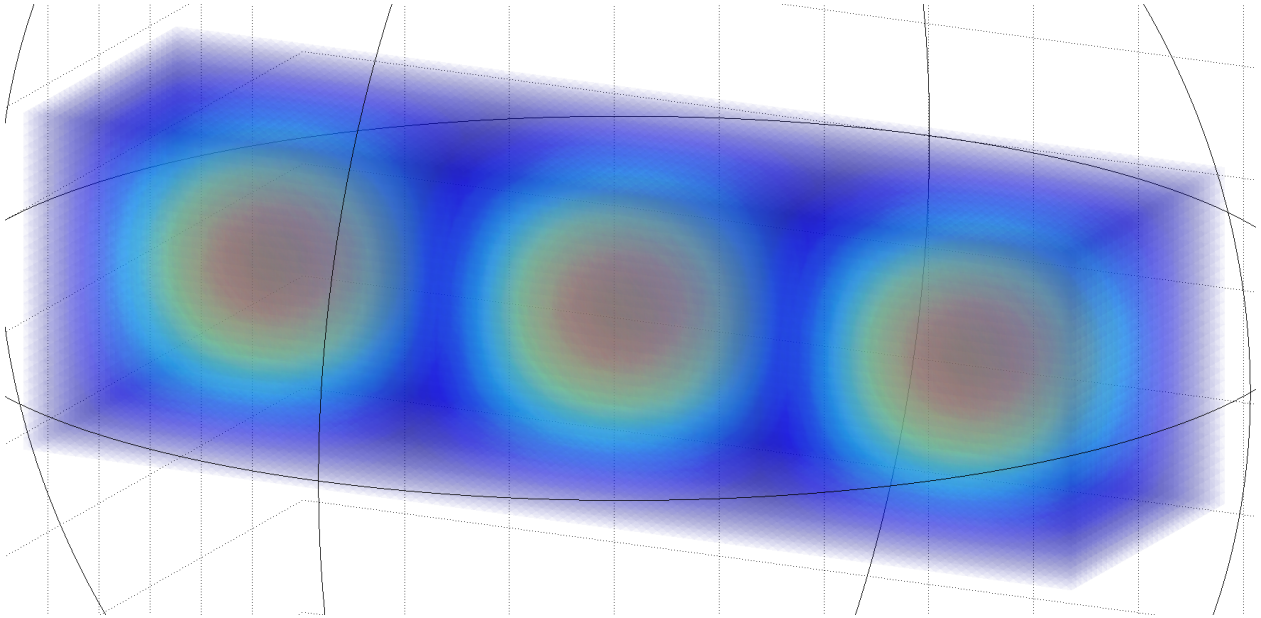


FIGURE : 3rd eigenfunction of (2) inside a box

Example 3 : Scattering by a (sound soft) sphere

- Acoustic scattering equation for domain Ω with boundary Γ :

$$\begin{cases} -(\Delta p + k^2 p) = 0 \quad \text{in} \quad \Omega^+, \\ \lim_{r \to +\infty} r\,(\partial_r p + ikp) = 0, \\ p = -p_{inc} \quad \text{in} \quad \Gamma. \end{cases}$$

- Incident plane wave with direction of propagation $\mathbf{d}$:

$$p_{inc}(\mathbf{x}) = \exp(ik\mathbf{x} \cdot \mathbf{d}).$$

- Galerkin formulation for a Dirichlet problem :

$$\int_\Gamma \int_\Gamma \phi(\mathbf{x}) G(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y})\, dx\, dy = -\int_\Gamma \phi(\mathbf{x}) p_{inc}(\mathbf{x})\, dx\,. \qquad (3)$$

- $G(\mathbf{x}, \mathbf{y})$ is the Helmholtz green kernel :

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

Example 3 : Scattering by a (sound soft) sphere

```
% Parameters
N  = 10^4; tol = 10^-3;
X0 = [0 0 -1];
k  = 2*pi*1000/340

% Spherical mesh
mesh = mshSphere(N,1);
S2   = dom(mesh,3);

% Incident wave
PW = @(X)exp(1i*k*X*X0');

% Green kernel
Gxy = @(X,Y) ...
femGreenKernel(X,Y,...
'[exp(ikr)/r]',k);

% Finite element space
Vh = fem(mesh,'P1');

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

% Regularization
LHS = LHS + 1/(4*pi)* ...
regularize(S2,S2,Vh,...
'[1/r]',Vh);

% Right hand side
RHS = integral(S2,Vh,PW);

% Solve linear system
lambda = LHS \ RHS;
```

Example 3 : Scattering by a (sound soft) sphere

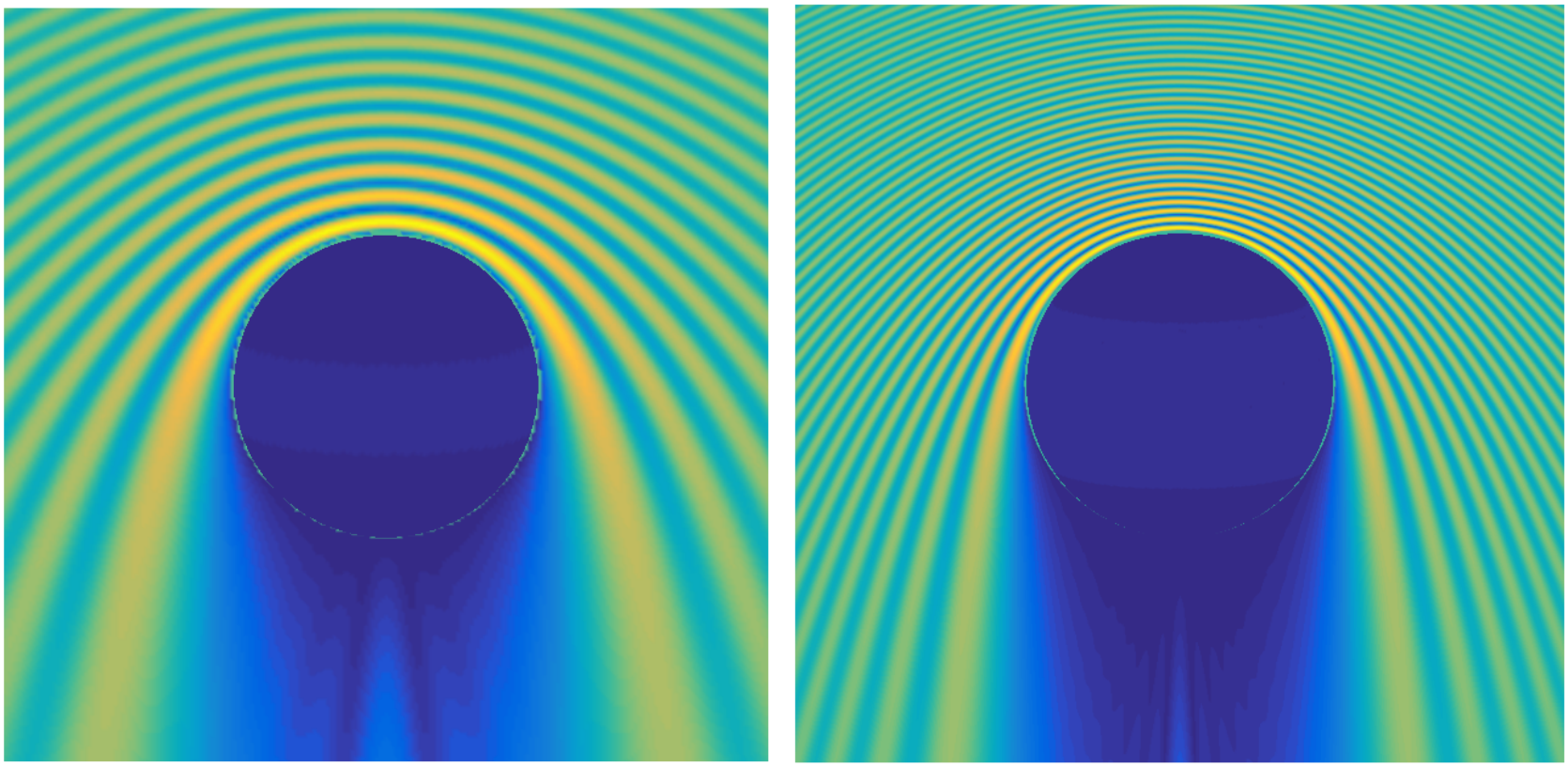
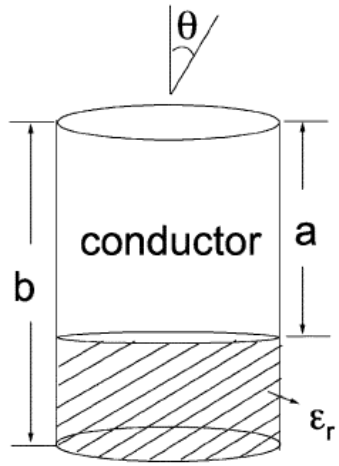


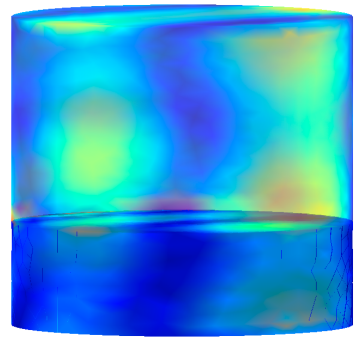
FIGURE : Plane wave scattering by a soft sphere, $\mathcal{H}$-Matrix exact solver. **Left :** 10 000 dof and 1 kHz. **Right :** 100 000 dof and 3kHz.

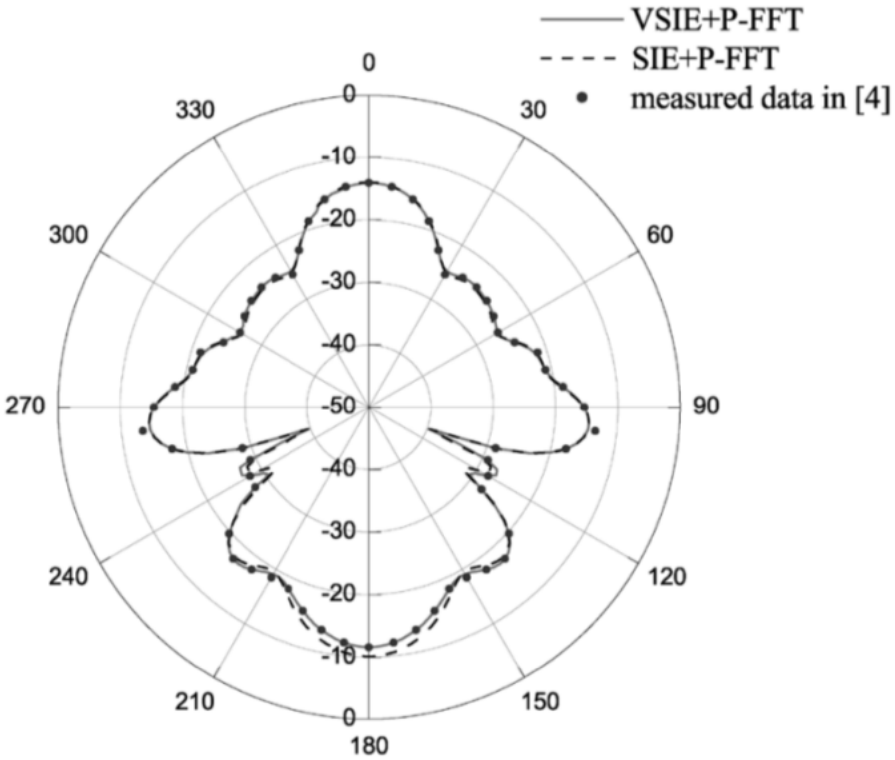
Example 4 : Electromagnetic scattering PEC & Dielectric

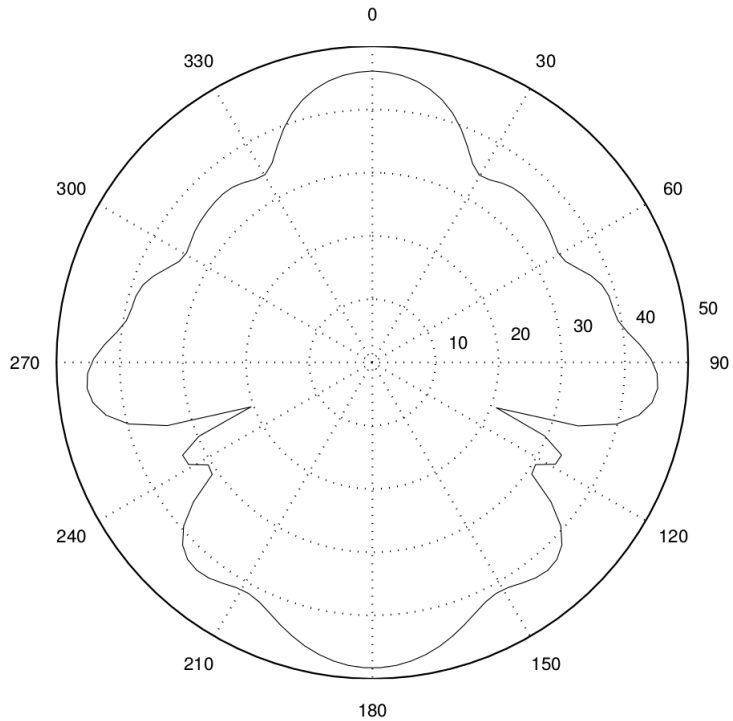
joint work with François Alouges

Mawxell equation
FEM/BEM coupling (PEC & diel.)
10.000 dof at 3 GHz
2D *RWG* and 3D *NED* FE









1

1. *A Fast Volume-Surface Integral Equation Solver Scattering From Composite Conducting-Dielectric Objects.*

Example 5 : Scattering by a (sound hard) head

Helmholtz equation

Neumann boundary condition

20.000 dof at 8 kHz

P^1 finite element

10^{-3} compression accuracy

$\mathcal{H}$-Matrix exact solver

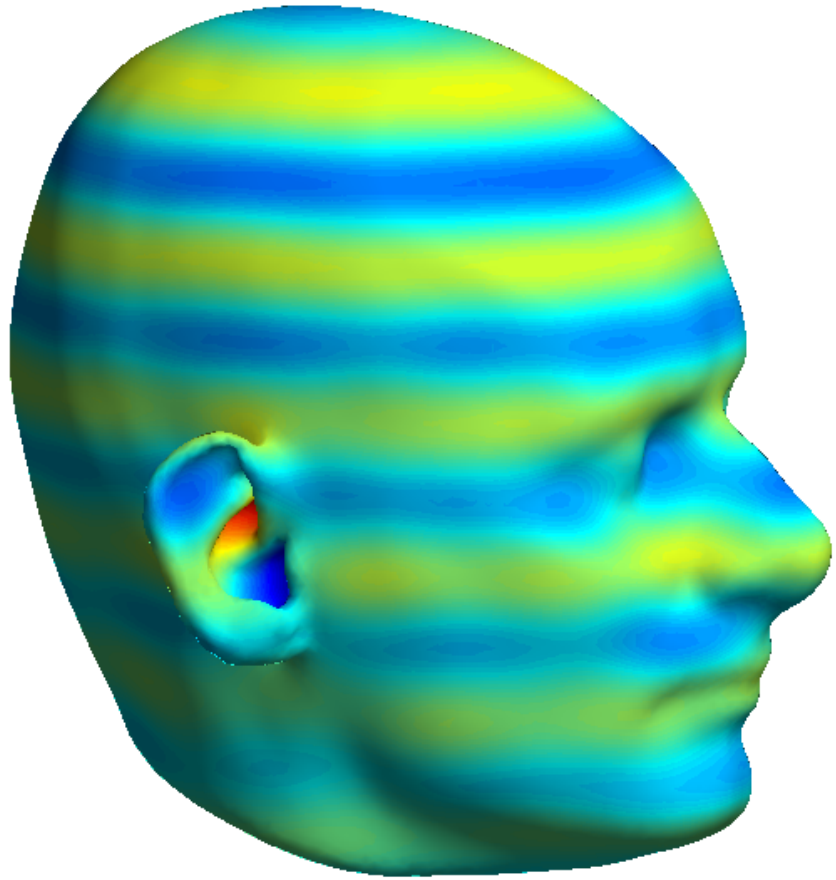


FIGURE : Acoustic scatering by human head, mesh from SYMARE database.

Example 6 : BeTSSI

Helmholtz equation
Neumann boundary condition
100.000 dof at 200 Hz

P^1 finite element
10^{-3} compression accuracy
$\mathcal{H}$-Matrix exact solver and radiation

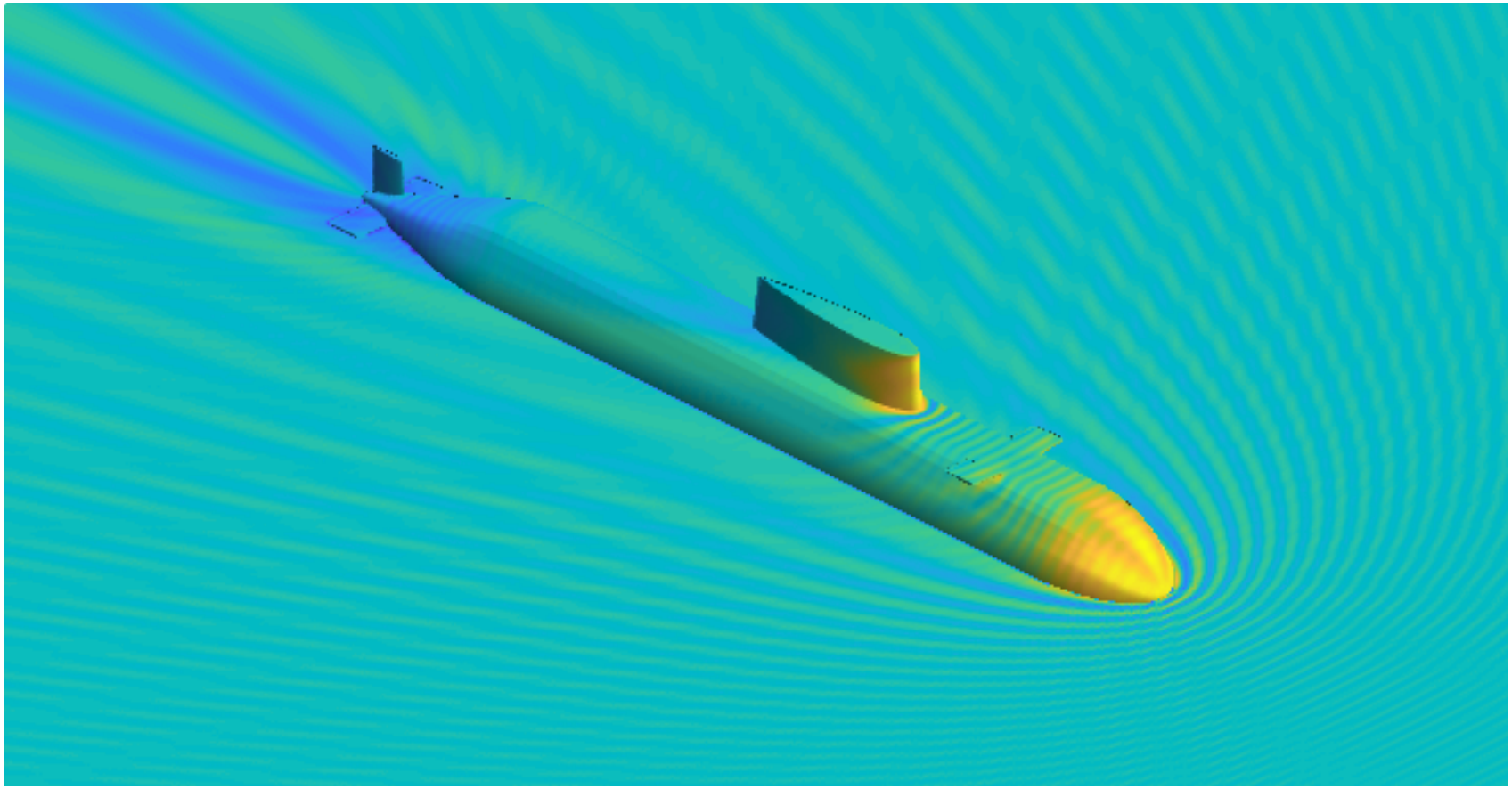


FIGURE : Acoustic scattering by a submarin, workshop BeTSSI, mesh from ESI-group.

Example 7 : Electromagnetic scattering (workshop ISAE)

Maxwell equation

PEC boundary condition, CFIE

300.000 dof at 2.5 GHz

RWG finite element

10^{-3} compression accuracy

$\mathcal{H}$-Matrix exact solver

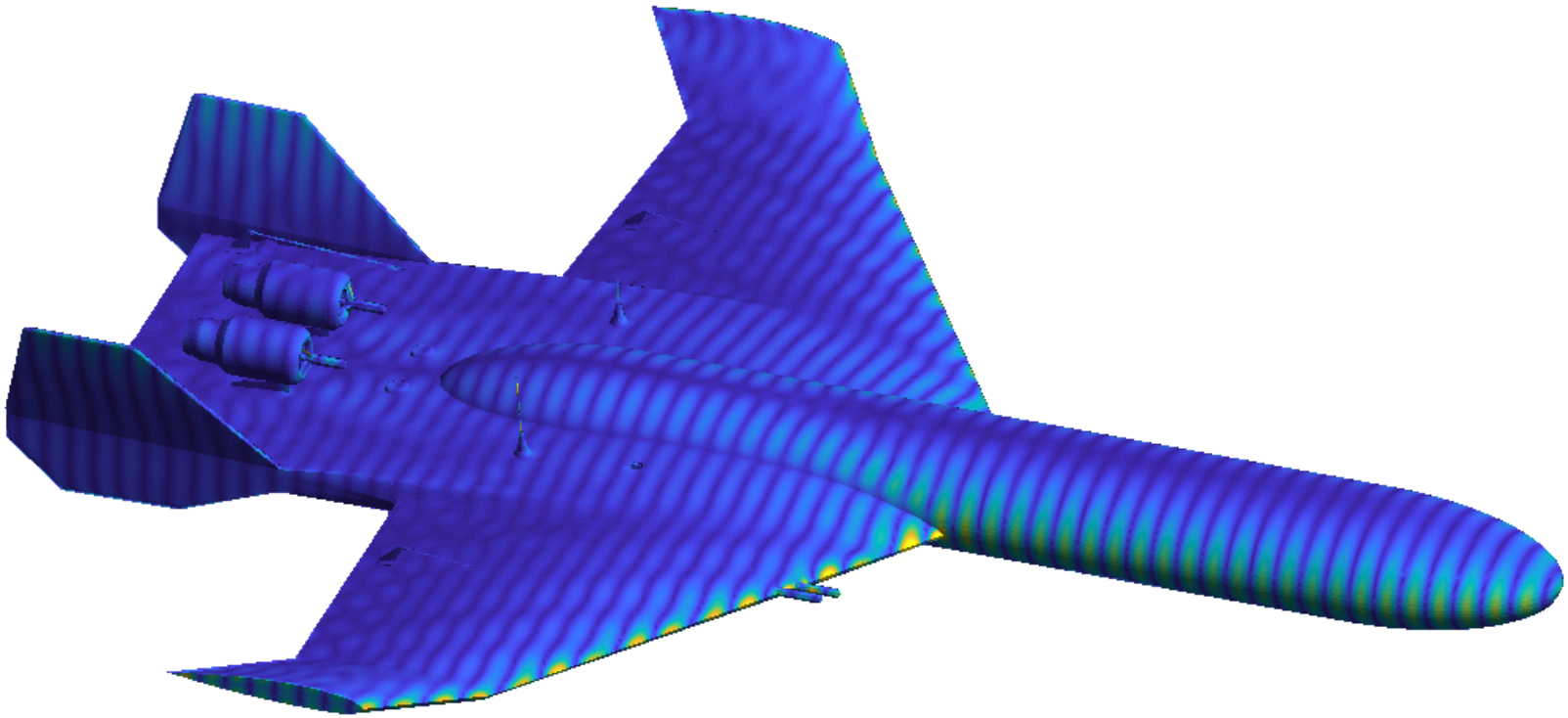


FIGURE : Electromagnetic scattering by an UAV, workshop ISAE, mesh from Onera.

Licensed add-on for GYPSILAB

Open source part of GYPSILAB is accompanied by **licensed add-on**, also composed of object-oriented libraries :

- **libHbm :** Block matrix computation and algebra.
  Domain Decomposition Method, $\mathcal{H}$-Matrix Parallelization, multi-physic coupling, out-of core computation, etc.

- **libFfm :** Fast and Furious Method.
  High Performance Computing for very large BEM problems.

Example 0 : Discrete convolution with Laplace kernel

Context : For all particles $(\mathbf{x}_i)_{i\in[1,N]}$ and $(\mathbf{y}_j)_{j\in[1,N]}$ in $\mathbb{R}^3$:

$$u(\mathbf{x}_i) = \sum_{j=1}^{N} \frac{1}{4\pi|\mathbf{x}_i - \mathbf{y}_j|} v(\mathbf{y}_j)$$

Computer : 12 cores. at 2.7 GHz, 256 Go de ram, Matlab R2016a.

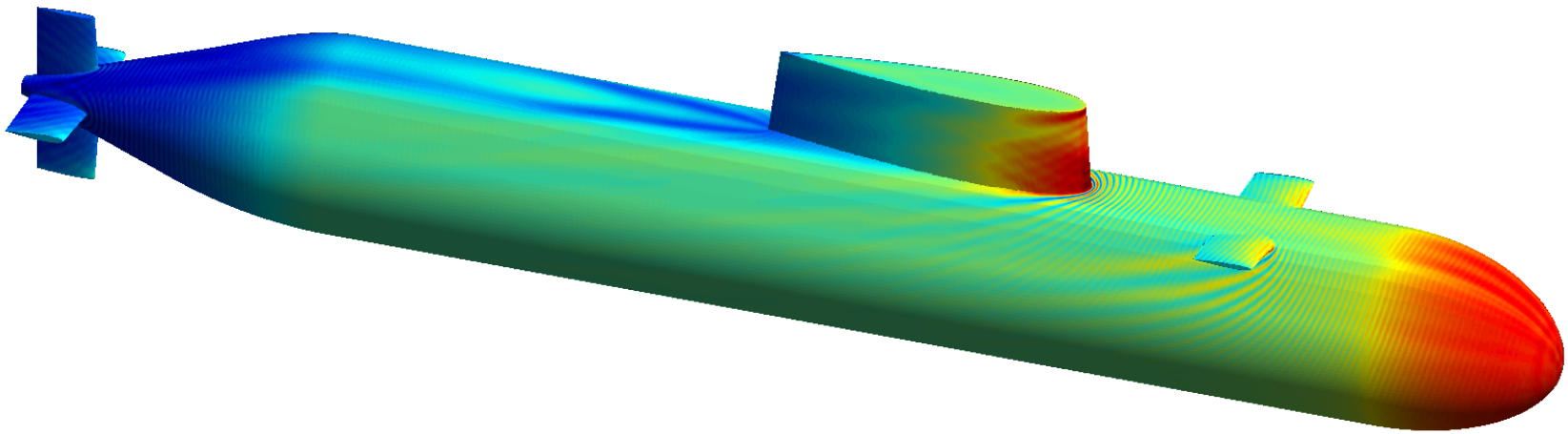
N	Time 1 core (s)	Time 12 cores (s)	Error L2	Mem
10^4	2.04	9.08	$8.03\ 10^{-5}$	1 Mo
10^5	9.30	17.1	$1.34\ 10^{-4}$	10 Mo
10^6	87.8	33.4	$1.35\ 10^{-4}$	100 Mo
10^7	1063	169	$1.98\ 10^{-4}$	1 Go
10^8	-	1499	$1.81\ 10^{-4}$	10 Go
10^9	-	11340	$3.11\ 10^{-4}$	100 Go

Fast and Furious product with $\epsilon = 10^{-3}$ (quasi-linear)

Exemple 8 : BeTSSI high frequency

Helmholtz equation
Neumann boundary condition
1.000.000 dof at 650 Hz

P^1 finite element
10^{-3} compression accuracy
Block $\mathcal{H}$-Matrix exact solver



Exemple 8 : BeTSSI high frequency

Helmholtz equation

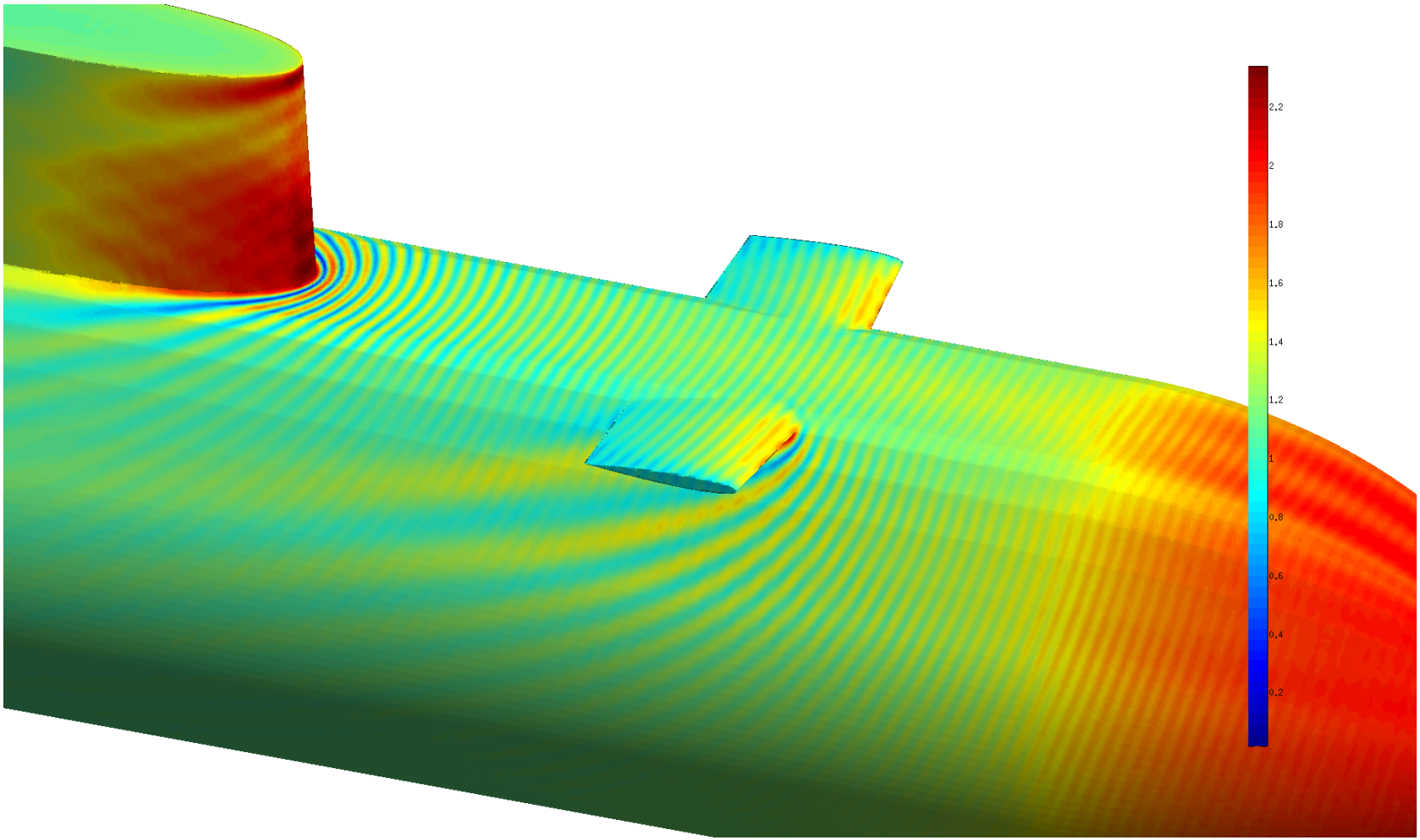
Neumann boundary condition

1.000.000 dof at 650 Hz

P^1 finite element

10^{-3} compression accuracy

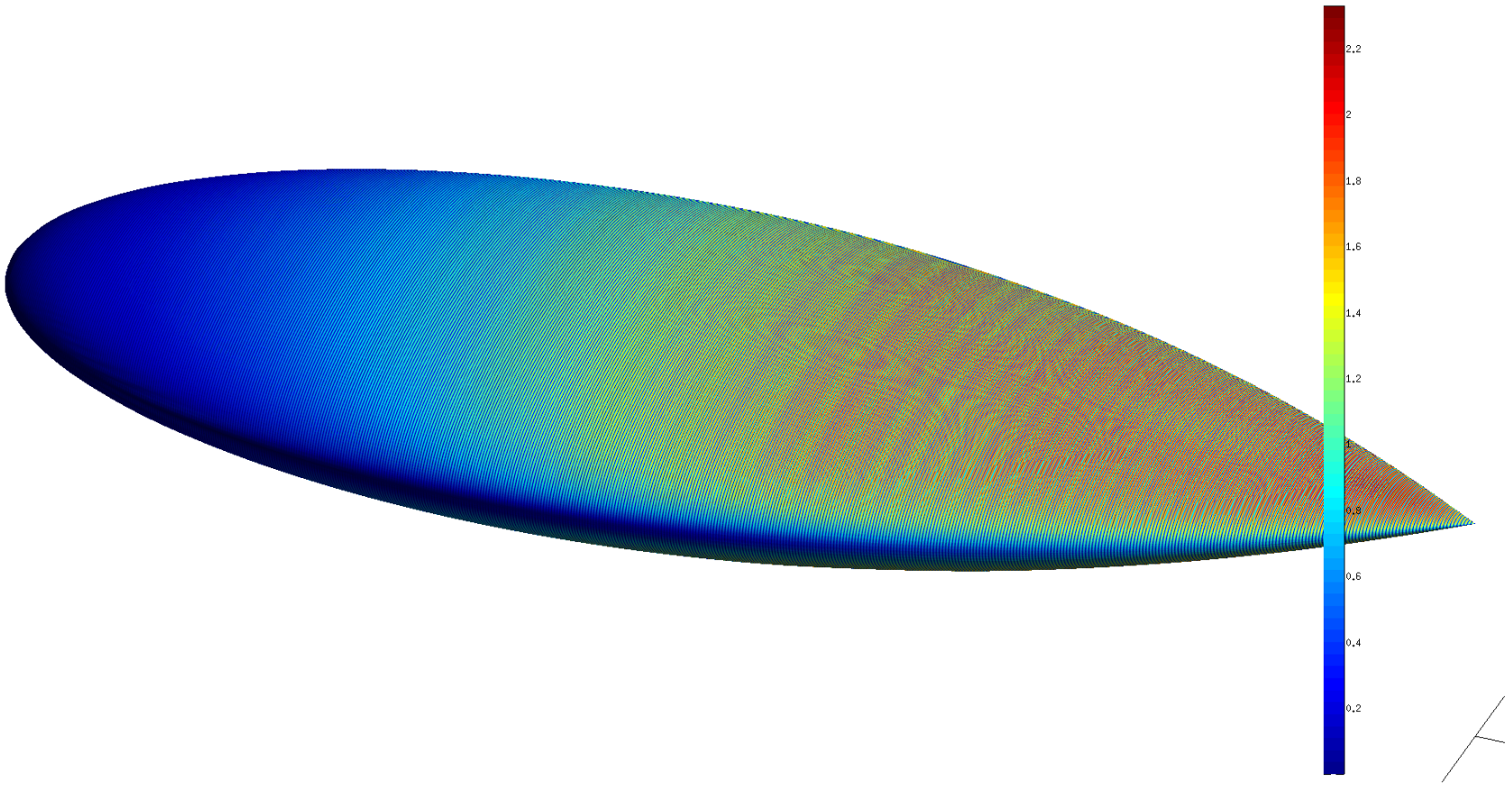
Block $\mathcal{H}$-Matrix exact solver



Example 9 : NASA almond

Maxwell equation
PEC boundary condition, CFIE
15.000.000 dof at 32 GHz

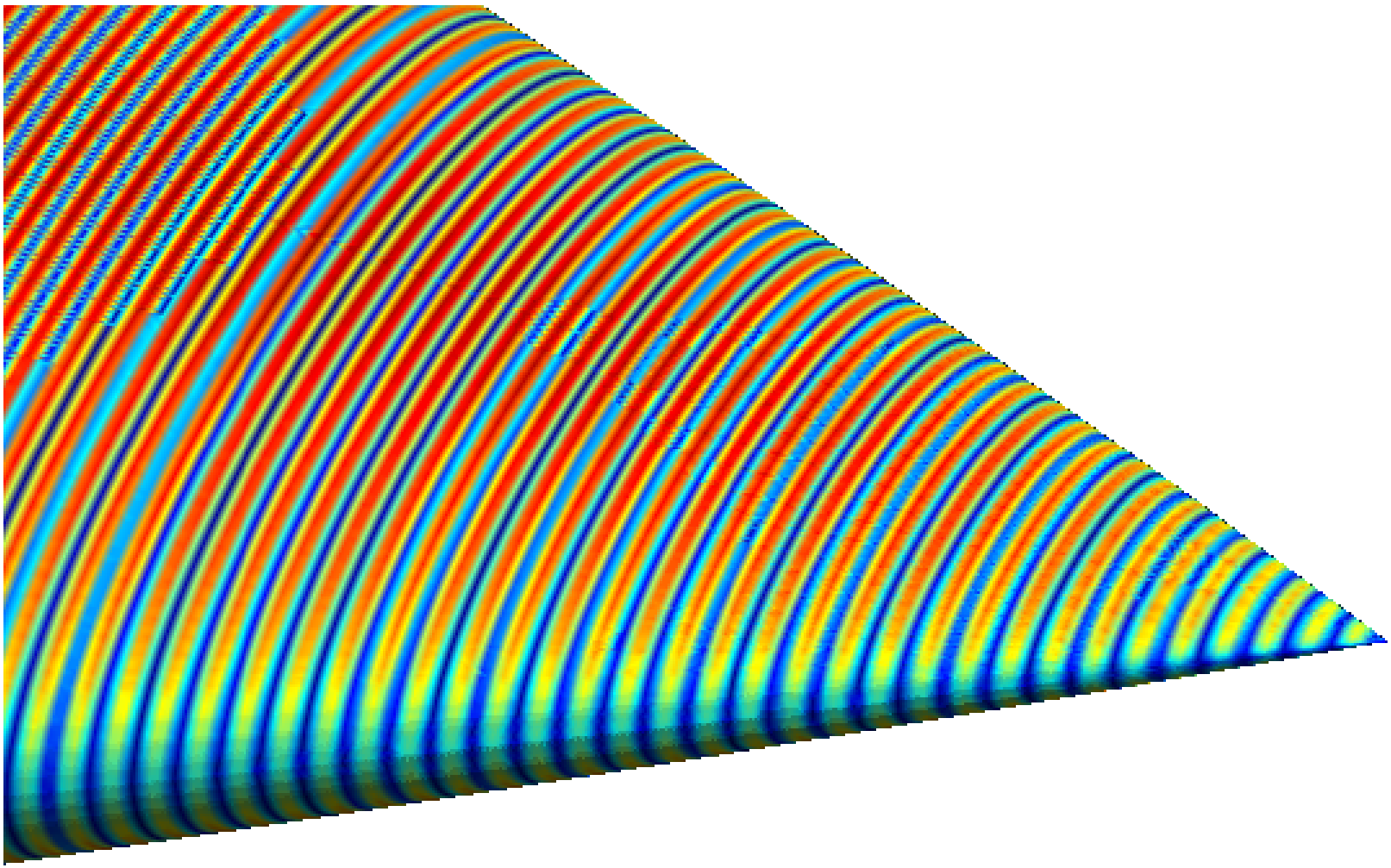
RWG finite element
10^{-3} compression accuracy
Preconditionned FFM iterative solver



Example 9 : NASA almond

Maxwell equation
PEC boundary condition, CFIE
15.000.000 dof at 32 GHz

RWG finite element
10^{-3} compression accuracy
Preconditionned FFM iterative solver

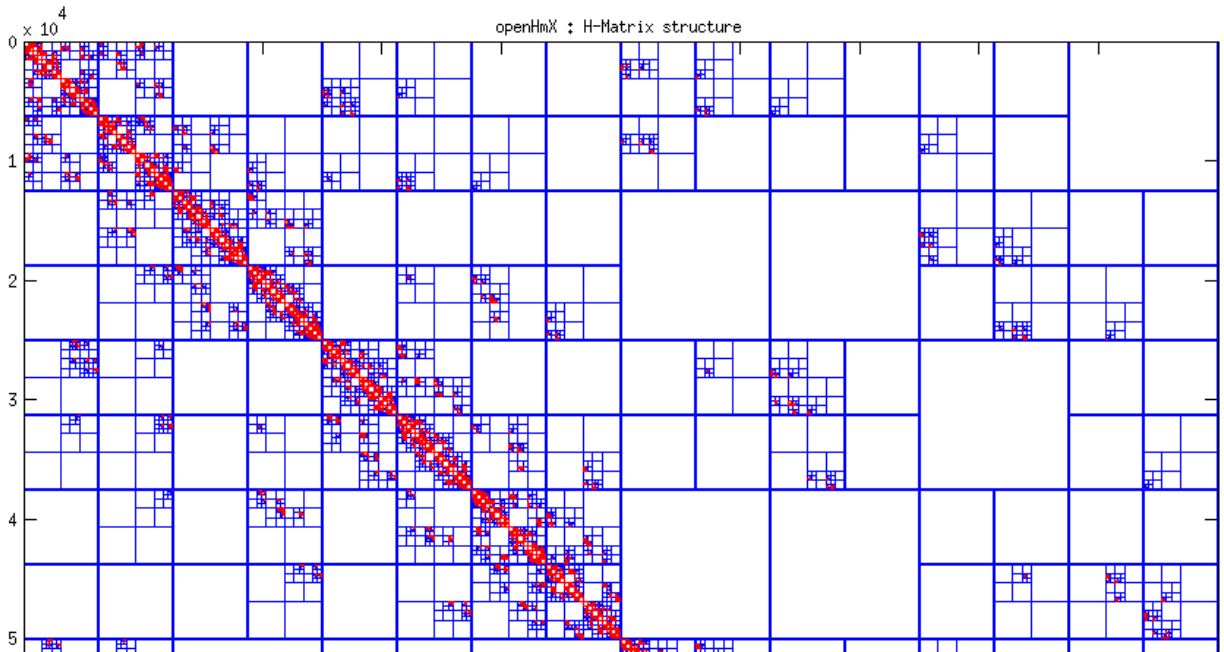


Next steps

- Coupling FEM/BEM/Ray-tracing,
- Add others numerical methods (FDTD,FVM,etc.)
- Mesh generation (2D, 3D, multi-d),
- Holidays.

Feel free to download GYPSILAB here :

https ://github.com/matthieuaussal/gypsilab



Few references

- Aussal, M. (2014) *Méthodes numériques pour la spatialisation sonore, de la simulation à la synthèse binaurale*. PhD thesis.
- Borm, S., Grasedyck, L., & Hackbusch, W. (2003). *Introduction to hierarchical matrices with applications*. Engineering analysis with boundary elements, 27(5), 405-422.
- Greengard, L., & Rokhlin, V. (1987). *A fast algorithm for particle simulations*. Journal of computational physics, 73(2), 325-348.
- Lizé, B. (2014). *Résolution directe rapide pour les éléments finis de frontière en électromagnétisme et acoustique. Parallélisme et applications industrielles*. Doctoral dissertation, Université Paris XIII.
- Terrasse, I., & Abboud, T. (2007). *Modélisation des phénomenes de propagation d'ondes*. Ecole Polytechnique.